Siemens Business Services GmbH & Co. OHG

Softwareprodukt

CIS

Version 12.0

Manual 2

DIENSTPROGRAMME

Ausgabe Oktober 1994

... und Schulung?

Zu dem nachstehend beschriebenen Produkt, bieten
wir Kurse in unserem Training-Center in München an.

Telefon/Fax (089) 636-4 89 99/-4 89 70

Siemens Nixdorf Training Center
Otto-Hahn-Ring 6 (Perlach)
81739 München

Briefadresse: 81730 München

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwendung und
Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden.
Zuwiderhandlungen verpflichten zu Schadenersatz.
Alle Rechte vorbehalten, insbesondere für den Fall der Patentierung oder GM-Eintragung.

Liefermöglichkeiten und technische Änderungen vorbehalten.

Copyright © Siemens Business Services GmbH & Co. OHG 1994
Alle Rechte vorbehalten.

Herausgegeben von

Siemens Business Services GmbH & Co. OHG
Professional Services, Region Süd
Berliner Straße 95
80805 München

Inhaltsverzeichnis

1 Einführung 7
2 CISGEN 9
2.1 Funktionsumfang 9
2.2 Bedienung 10
2.3 CISGEN - Funktionen 11
2.3.1 Kommandoklassen und Datenflußdiagramme 11
2.3.2 CISGEN - Kommandos 18
2.4 Satzbeschreibung - Datenbankdefinition 82
2.4.1 A-Segment der Satzbeschreibung 83
2.4.2 D-Segment der Satzbeschreibung 94
2.4.3 E-Segment der Satzbeschreibung 99
2.4.4 Beispiele für Satzbeschreibungen 100
2.5 Transformationsbeschreibung 107
2.5.1 A-Segment der Transformationsbeschreibung 107
2.5.2 D-Segment der Transformationsbeschreibung 108
2.5.3 E-Segment der Transformationsbeschreibung 113
2.5.4 F-Segment der Transformationsbeschreibung 115
2.5.5 Beispiele für Transformationsbeschreibungen 116
2.6 Bildbeschreibung 123
2.7 Bildschirmmasken 125
2.7.1 Aufbau der Bildschirmmaske 125
2.7.2 Hinweise zum Format der Bildschirmmaske 132
2.7.3 Beispiel einer Bildschirmmaske 133
2.8 Verwalten der Datenbankdefinitionen und Bildschirmmasken 134
2.8.1 Datenbeschreibungen und Bildschirmmasken in die DABEL aufnehmen 134
2.8.2 Datenbeschreibungen und Bildschirmmasken aus der DABEL regenerieren 134
2.8.3 Datenbeschreibungen und Bildschirmmasken definieren, duplizieren oder löschen 135
3 CISLADF 137
3.1 Funktionsumfang 137
3.2 Datenflußdiagramme 138
3.2.1 Datenfluß beim Laden der Hauptdatei 138
3.2.2 Datenfluß beim Laden der Verweisdatei 139
3.2.3 Datenfluß beim Laden der Haupt- und Verweisdatei 140
3.3 Hauptdatei laden 141
3.4 Verweisdatei laden 144
3.5 Hauptdatei und Verweisdatei gemeinsam laden 146
4 CISDBH 151
4.1 Funktionsumfang 151
4.2 Bedienung 152
4.2.1 Einrichten von CISDBH 152
4.2.2 Start des CISDBH-Prozesses 153
4.2.3 Parameter für Ablaufumgebung von CISDBH 154
4.2.4 Bedienung während des Betriebes 165
4.2.5 Beendigung von CISDBH 173
4.2.6 Dateinamen 174
4.3 Kommunikation zu CISDBH 175
4.3.1 Allgemeines 175
4.3.2 Benützung des Memory-Pools 175
4.3.3 Berechnen der Memory-Pool Größe 176
4.4 Datenflußpläne von CISDBH 177

4.5 Aufruf von CISDBH (über CISCON) ... 182
4.5.1 Betriebsarten ... 182
4.5.2 Parameter für CISCON ... 186
4.5.3 Zusammenwirken der CISDBH- und der CISCON-Parameter (Verteilregel) ... 191
4.6 Beispiele von Parametern ... 195
5 CISKOOR ... 199
5.1 Funktionsumfang ... 199
5.2 Datenflußplan von CISKOOR ... 200
5.3 Dateien ... 201
5.4 Einrichten von CISKOOR ... 202
5.5 Start des CISKOOR-Prozesses ... 203
5.5.1 Startprozedur ... 203
5.5.2 Multi-Task-Betrieb ... 204
5.5.3 Mehrere CISKOOR ... 204
5.5.4 Kaltstart, Warmstart und fehlertoleranter Warmstart ... 205
5.6 Verbindung CIS - CISKOOR ... 206
5.7 Beenden von CISKOOR ... 207
5.8 Benützte Schalter ... 207
5.9 Memory-Pool ... 208
5.9.1 Allgemeines ... 208
5.9.2 Benützung des Memory-Pools ... 208
5.9.3 Berechnen der Memory-Pool Größe ... 209
5.9.4 Anzahl der CIS-Tasks ... 210
5.10 Bemerkungen zum 2-Phasen-Commit-Protokoll ... 211
5.10.1 Allgemeines ... 211
5.10.2 Ablauf des Protokolls ... 211
5.11 Bemerkungen zum Warmstart ... 214
5.11.1 Offene Transaktion ... 214
5.11.2 Transaktion im Zustand PTC ... 214
5.11.3 Manuelle Eingriffe ... 214
5.11.4 Stati ... 214
5.11.5 PTC Zustände ... 214
5.12 Besonderheiten bei Inlinked CISKOOR ... 215
5.13 Parameter für CISKOOR ... 216
5.14 Bedienung während des Betriebs ... 228
6 CISINIT ... 241
6.1 Funktionsumfang ... 241
6.2 Bedienung ... 241
6.3 Steueranweisungen ... 241
6.4 Datenflußdiagramm ... 242
6.5 Dateinamen ... 243
6.6 Beispiel für den Ablauf von CISINIT ... 244
7 CISRA ... 245
7.1 Funktionsumfang ... 245
7.2 Allgemeiner Ablauf ... 245
7.3 Datenflußplan von CISRA ... 246
7.4 Bedienung ... 247
7.4.1 Steueranweisungen ... 247
7.4.2 Meldungen ... 248
7.4.3 Dateinamen ... 249
8 CISPROT ... 253
8.1 Funktionsumfang ... 253
8.2 Datenflußplan von CISPROT ... 253
8.3 Bedienung ... 254
8.3.1 Steueranweisungen ... 254
8.3.2 Selektionen ... 255
8.3.3 Dateinamen ... 257

9 CISSERV 259
9.1 Funktionsumfang 259
9.2 Datenfluß 260
9.3 Bedienung 261
9.3.1 Einrichten von CISSERV 261
9.3.2 Start des CISSERV-Prozesses 261
9.3.3 Parameter für CISSERV 262
9.3.4 Bedienung während des Betriebs 267
9.3.5 Beendigung von CISSERV 269
10 CISDICT 271
10.1 Funktionsumfang 271
10.2 Auskunftsfunktionen 272
10.2.1 Auswahl der Auskunftsfunktionen 272
10.2.2 Allgemeine Übersichten 273
10.2.3 Gezielte Auswertung über HD 275
10.2.4 Gezielte Auswertung über Satzbeschreibungsnamen 275
10.2.5 Gezielte Auswertung über die Transformations - oder Bildbeschreibung 276
10.2.6 Gezielte Auswertung über Maskennamen 276
10.2.7 Gezielte Auswertung über Feldbezeichnung 277
10.2.8 Übersicht im Dateiformat 279

1 Einführung

Das komplette Datenbanksystem CIS besteht aus dem Data-Base-Handler CIS und Dienstprogrammen, die für die Definition, den Aufbau und die Wartung der Datenbank notwendig sind.

In dieser Beschreibung werden die Funktion und die Bedienung dieser Dienstprogramme erläutert. Soweit sinnvoll und notwendig werden getestete Beispiele dargestellt.

Tabellarische Übersicht der Dienstprogramme:

CISGEN	Datenbeschreibungsgenerator
CISLADF	Ladeprogramm zum Aufbau der Haupt- und Verweisdatei.
CISDBH	Unabhängiger CIS Data-Base-Handler.
CISKOOR	Koordinationsprozeß der Datensicherung.
CISINIT	Initialisierung der Datensicherungsdateien.
CISRA	Wiederherstellung der CIS-Dateien mit After-Image-Sätzen.
CISPROT	Ausdruck der After-Image-Datei und der Protokolldatei.
CISSERV	Client - Server
CISDICT	Auswertung Datenbeschreibungsdatei.

2 CISGEN

2.1 Funktionsumfang

Mit CISGEN werden die Datenbank selbst, die Zugriffswege, der Datenschutz und bei Bedarf Transformationen sowie Bildschirmformate und Listenformate definiert. Alle diese Informationen und die CIS-Texte werden in der Datenbeschreibungsdatei DABEL gespeichert. CISGEN verwaltet also die Datenbeschreibungsdatei.

Die in CISGEN implementierten Funktionen lassen sich in folgende Klassen einteilen:

Funktionsklasse	Funktionsbeschreibung
Einrichten	Erzeugung einer DABEL und Laden der Texte in die DABEL.
Generieren	Laden von Datenbeschreibungen und Masken in die DABEL.
Regenerieren	Lesen von Datenbeschreibungen und Masken aus der DABEL.
Definieren	Aufbauen von Datenbeschreibungen und Masken.

CISGEN wird ausgeliefert als:

- Modul
  Es kann von einem Programm aufgerufen werden.

- Phase
  Es kann im TIAM-Dialog oder Batch bedient werden.

Funktionsweise

Die Datenbeschreibungen und Masken werden im EDT-Speicher aufgebaut: (CISGEN benützt als Unterprogramm den EDT des BS2000)

- aus der DABEL (RGEND, RGENM)
- aus einer Datei (READ, GET)
- aus einer Fremdbibliothek (CDB, IDB, LDB, SDB)
- per Kommando (ADB, SEG)

In diesem Speicher können sie mit EDT-Anweisungen bearbeitet werden (EDT-L und EDT-F als Unterprogramm).

Aus diesem Speicher können sie auch in eine DABEL geladen werden (GEND, GENM, PUTD, PUTM).

2.2 Bedienung

Alle CIS-Meldungen und gültigen Kommandos werden in der jeweiligen Sprache in der DABEL abgelegt. Es gibt hierzu zwei Datenbeschreibungsgeneratoren:

- CISGEN (deutsch)

- CISGENE (englisch)

Ist bei einem Sprach- und/oder Versionswechsel eine Datenbeschreibungsdatei vorhanden, so werden automatisch beim Starten des Datenbeschreibungsgenerators die Texte in entsprechender Sprache in die Datenbeschreibungsdatei geladen. Heißt die Datenbeschreibungsdatei nicht DABEL, so muß sie vorher explizit mit

```
/SET-FILE-LINK FILE-NAME=Datenbeschreibungsdatei,LINK-NAME=DB
```

zugewiesen werden.

Ist bisher noch keine Datenbeschreibungsdatei vorhanden, so wird diese beim Aufruf des Datenbeschreibungsgenerators von diesem wie folgt standardmäßig angelegt:

Name	DABEL
KEYPOS	7
KEYLEN	6
BLKSIZE	(STD,3)
SPACE	(70,16)

Die entsprechenden Texte (CISGEN = deutsch, CISGENE = englisch) werden geladen.

Soll die Datei nicht standardmäßig eingerichtet werden, so muß der Benutzer vor Aufruf des Datenbeschreibungsgenerators ein CREATE-FILE Kommando geben und mit dem SET-FILE-LINK Kommando zuweisen:

```
/CREATE-FILE FILE-NAME=Datei,SUPPORT=PUBLIC-DISK(SPACE=RELATIVE
(PRIMARY-ALLOCATION=x,SECONDARY-ALLOCATION=y))

/SET-FILE-LINK FILE-NAME=Datei,LINK-NAME=DB,ACCESS-METHOD=ISAM
(KEY-LENGTH=6,KEY-POSITION=7),BUFFER-LENGTH=STD(SIZE=n)
```

CISGEN wird über SYSDTA bzw. von einem Benutzerprogramm gesteuert.

Sämtliche CIS-Texte können mit der LSTE-Anweisung in einer druckaufbereiteten Liste auf SYSLST ausgegeben werden.

Bedienung als Programm

Zuweisen der DABEL mit: `/SET-FILE-LINK FILE-NAME=Dabelname,LINK-NAME=DB`

Laden von CISGEN mit: `/START-PROGRAM FROM-FILE=CISGEN`

Wird kein SET-FILE-LINK Kommando gegeben, so wird als Dabelname DABEL benützt.

CISGEN gibt die Versionsmeldung aus und fordert dann Parameter an.

Bedienung als Modul

Die CISGEN-Schnittstelle ist im Manual-4 (SCHNITTSTELLEN) "CISGEN als Unterprogramm" beschrieben.

2.3 CISGEN - Funktionen

2.3.1 Kommandoklassen und Datenflußdiagramme

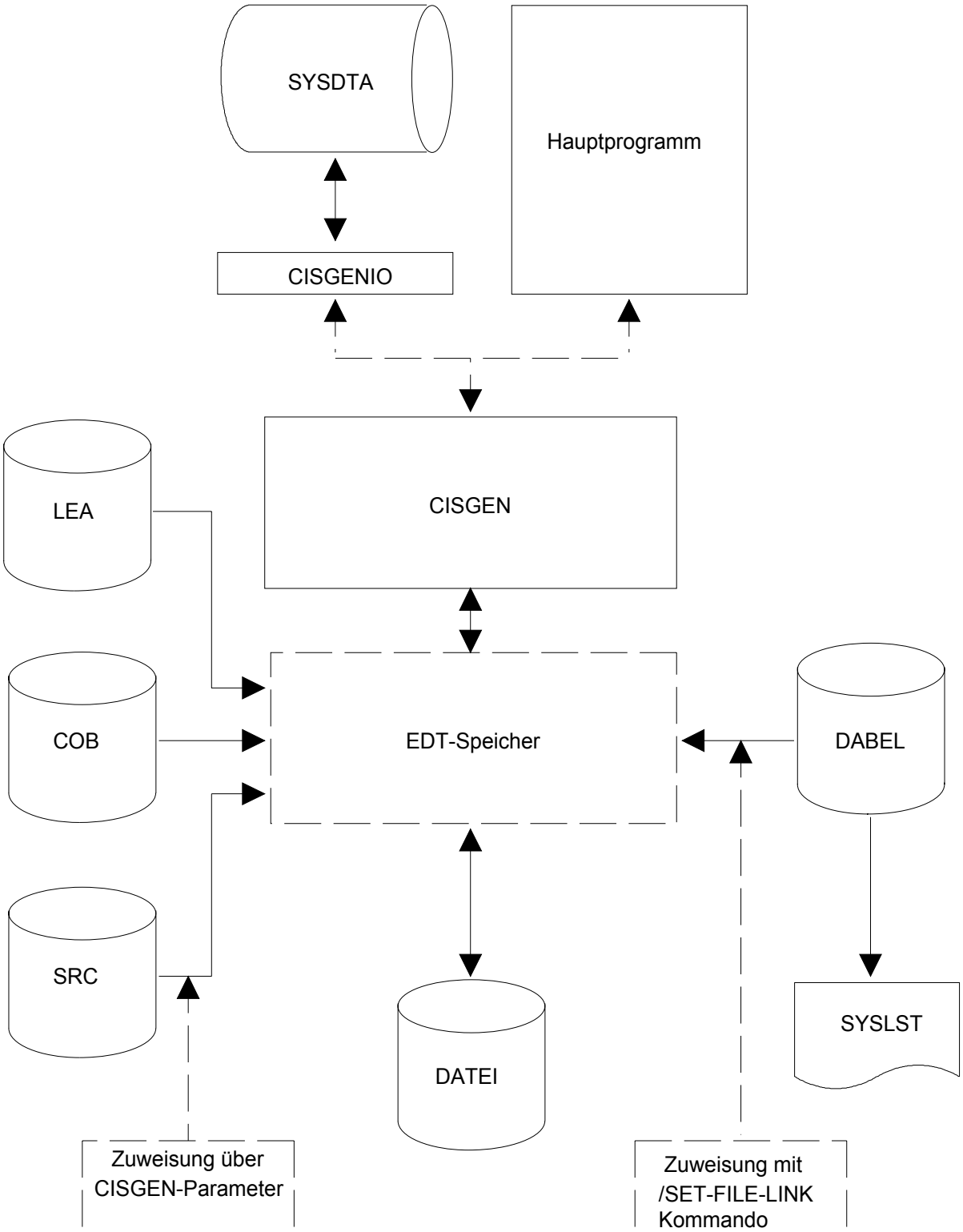
Kommandoklassen

Klasse	Kommando	Funktionsbeschreibung
Bibliotheken	COB LEA SRC	Coblur-Datei LEASY-Katalog LMS-Datei } eröffnen
	DELD DELM	Daten-beschreibungen Masken } löschen
	CINF DINF DINFD DINFM LINF SINF	DABEL LEASY-Katalog LMS-Datei } ausgeben des Inhaltsverzeichnisses auf SYSOUT
	LSTD LSTE LSTI LSTM	Datenbeschreibungen Fehlertexte Infotexte Masken } auflisten auf SYSLST
Speicher ↕ DABEL	RESET	EDT-Speicher löschen
	RGEND RGENM	Datenbeschreibungen Masken } regenerieren d.h. in EDT-Speicher schreiben
	GEND PUTD GENM PUTM	Datenbeschreibungen Masken } generieren d.h. in die DABEL schreiben
Speicher ↕ Datei	GET READ SAVE WRITE	ISAM - Datei lesen SAM - Datei lesen ISAM - Datei schreiben SAM - Datei schreiben

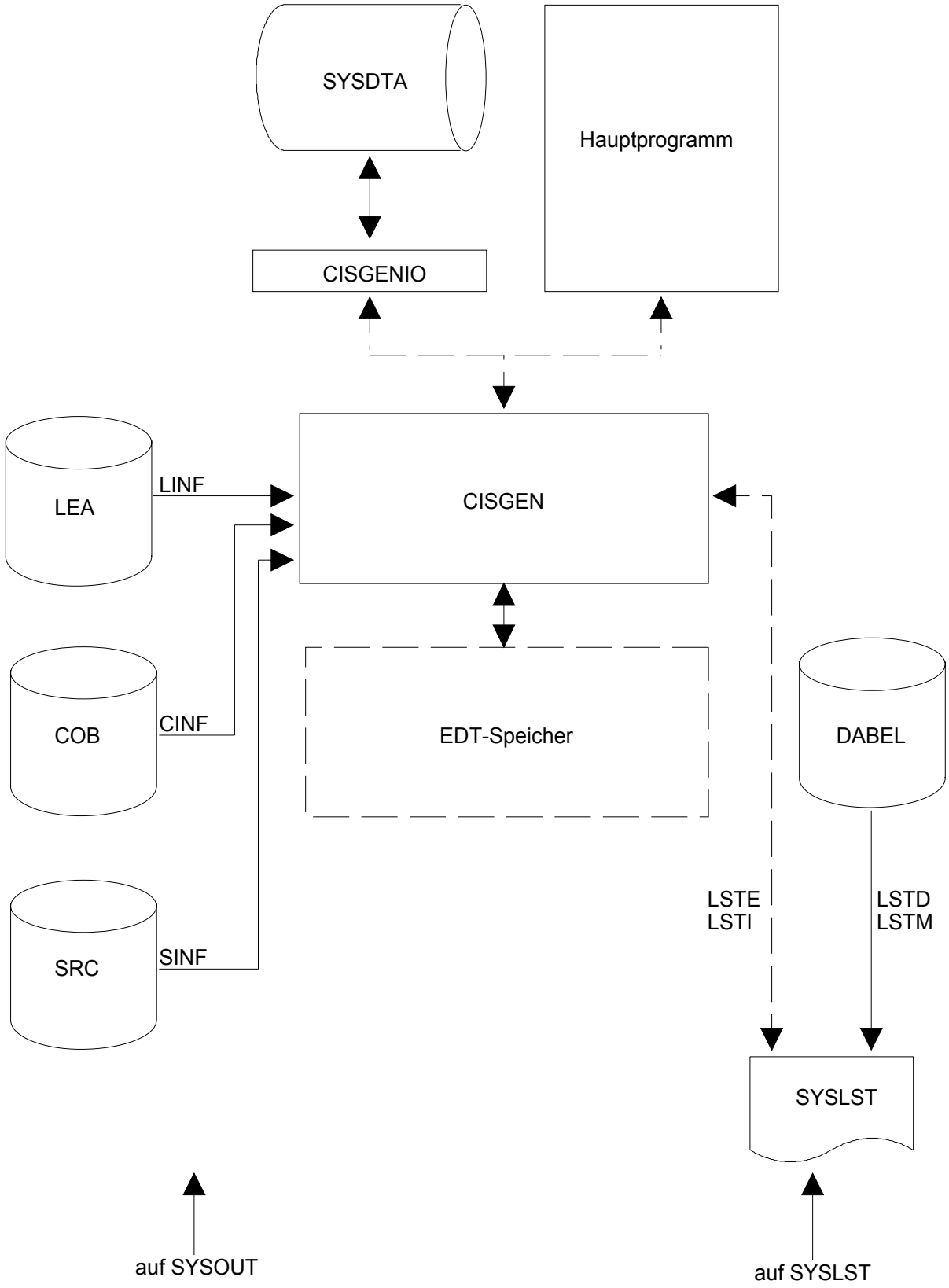
Klasse	Kommando	Funktionsbeschreibung
Fremdbibliothek ↓ Speicher	CDB IDB LDB SDB	aus { COBLUR-Datei / SAM-/ISAM-Datei / LEASY-Katalog / LMS-Datei } Datenbeschreibungen generieren
Programm ↕ Speicher	USETB UREAD UWRITE EDT	Lesezeiger auf Anfang setzen Satz von CISGEN lesen Satz in EDT-Speicher schreiben EDT-L Kommando absetzen
EDT	EDT EDTP EDTU EDTF	ruft { EDT-L / EDT-L mit @PRINT / EDT-L mit @UPDATE / EDT-F } auf
Direkteingaben in die DABEL	DEFD DEFM DEFT	Datenbeschreibungen / Masken } definieren Transformationsbeschreibung generieren
Sonstige	* ADB ERRPUT HALT NOP PROC RECF SEG SORT SYSDTA TERMINAL WHY	Kommentar A-Segment definieren Fehlerbehandlung Programm beenden Nulloperation Meldungen unterdrücken RECFORM für LEASY-Datei eingeben D und E Segment generieren Sortieren einer Datenbeschreibung nach Adressen EDT-Eingabe auf SYSDTA setzen EDT-Eingabe auf Terminal setzen Warum-Funktion

Datenflußdiagramme

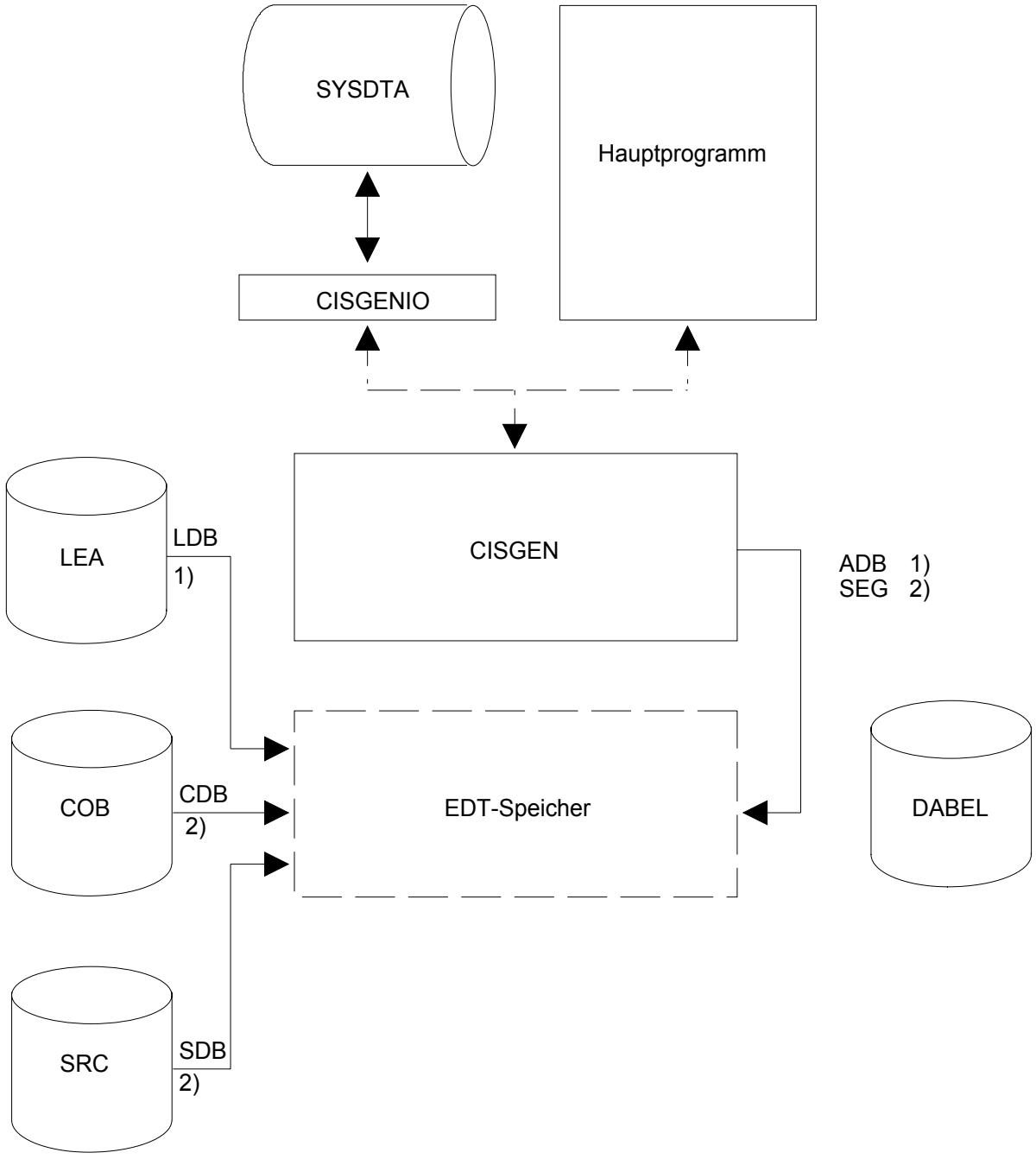
Gesamter Datenfluß



Inhaltsverzeichnisse auflisten



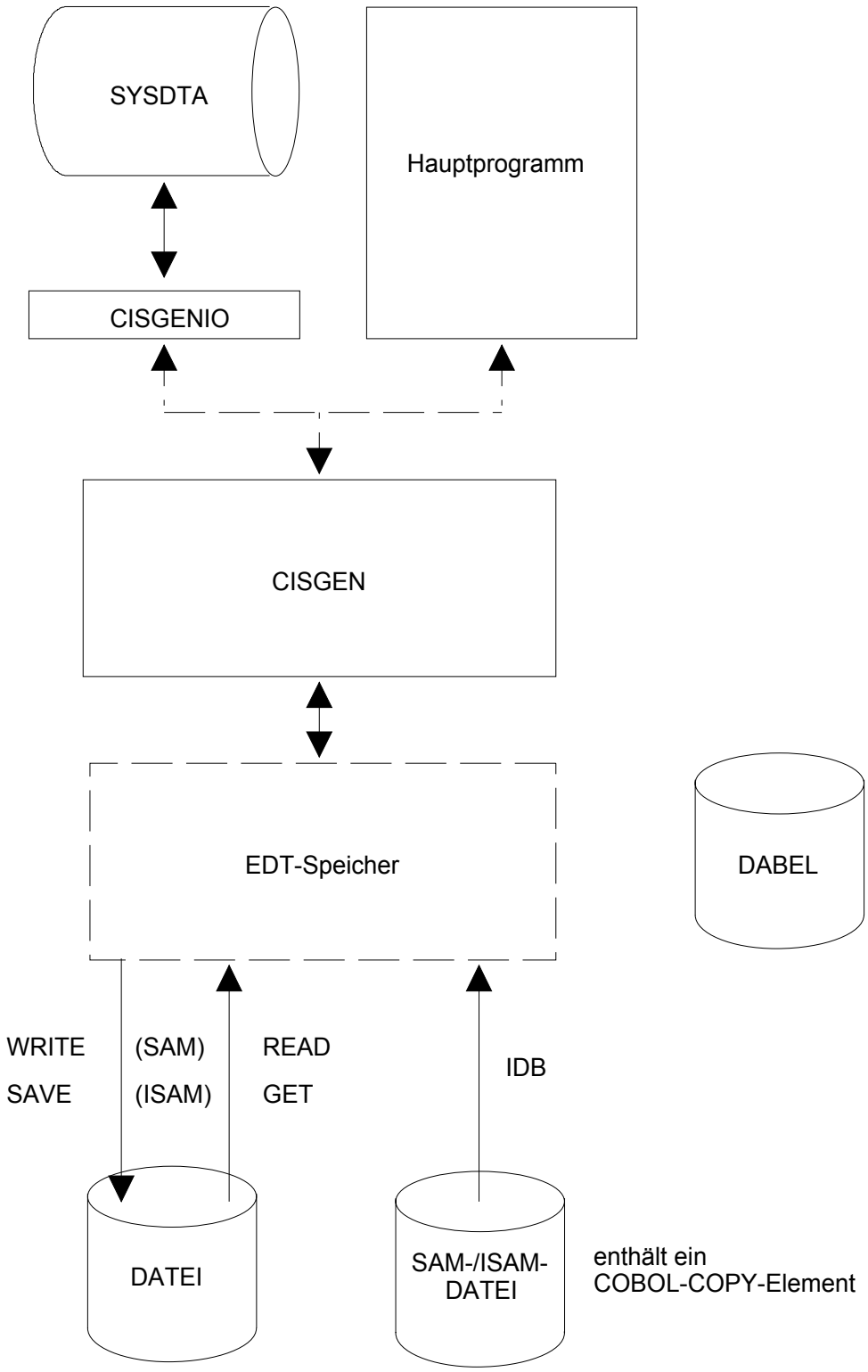
Bibliothekszugriffe



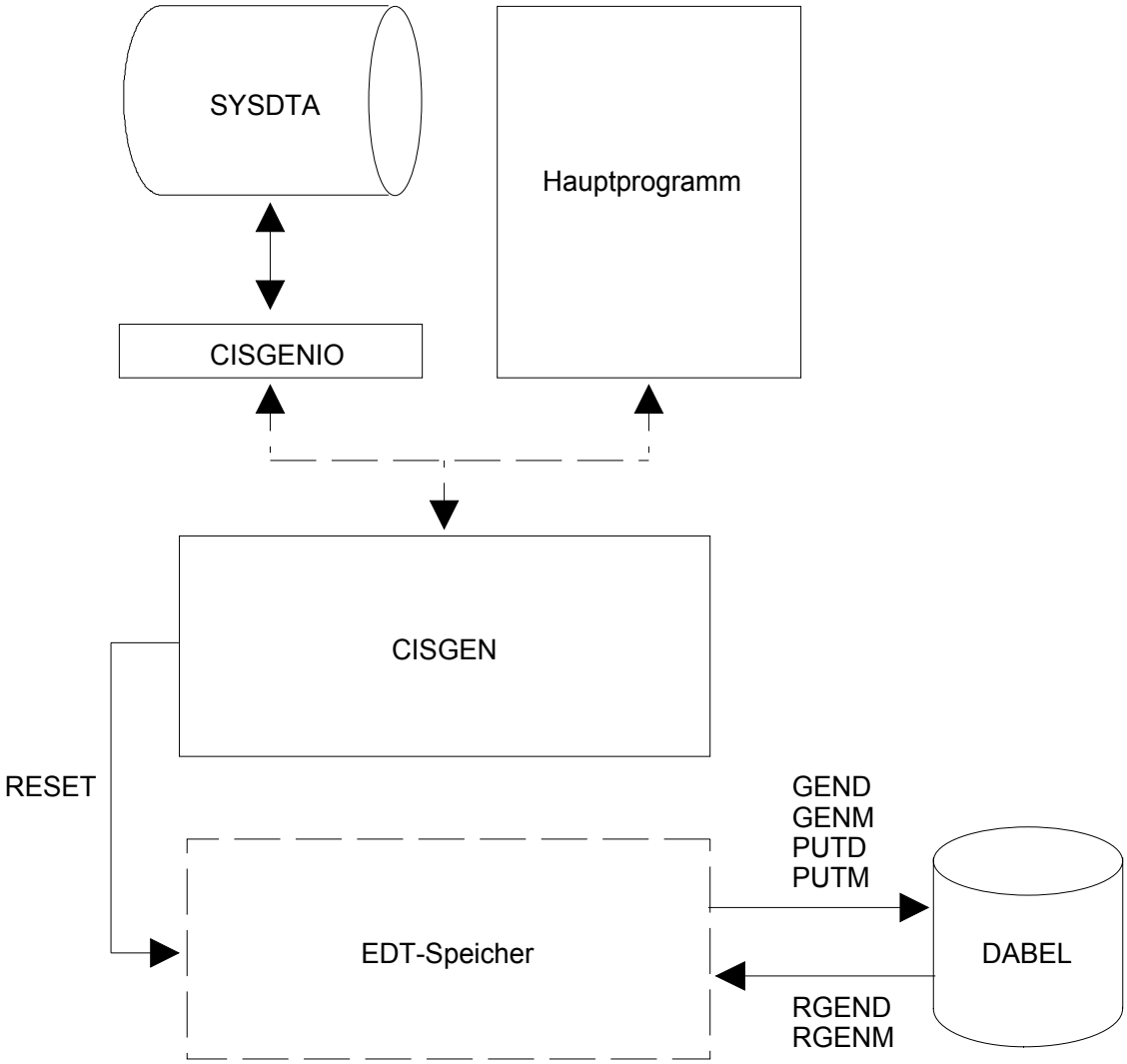
1) vorher wird der EDT-Speicher gelöscht

2) EDT-Speicher wird nicht gelöscht

Dateizugriffe



DABEL-Zugriffe



2.3.2 CISGEN - Kommandos

Tabellarische Übersicht der CISGEN - Kommandos

Kommando Erläuterung	Operand
* Kommentar	[_Kommentar]
ADB A-Segment definieren	
CDB Generieren aus COBLUR-Datei	_[Cn,]Name[,PRE=nn][,GRP=XXXX][,BEG=nn]
CINF Inhaltsverzeichnis aus COBLUR-Datei ausgeben	[_[Cn,]{Name/Teilname*}]
COB COBLUR-Datei eröffnen	_Dateiname
DEFD Datenbeschreibung definieren	
DEFM Masken definieren	
DEFT Transformationsbeschreibungen generieren	
DELD Datenbeschreibung löschen	_{Name/Teilname*/*}
DELM Masken löschen	_{Name/Teilname*/*}
DINF Inhaltsverzeichnis der DABEL ausgeben	_{Name/Teilname*}
DINFD Inhaltsverzeichnis der Datenbeschreibungen ausgeben	_{Name/Teilname*}
DINFM Inhaltsverzeichnis der Masken ausgeben	_{Name/Teilname*}
DSWITSCH Umschalten der DABEL	_Dateiname
EDT EDT aufrufen	
EDTF EDT aufrufen	
EDTP EDT aufrufen	
EDTU EDT aufrufen	

Kommando Erläuterung	Operand
EDT: EDT-L Kommando übergeben (CISGEN als Unterprogramm)	
ERRPUT Fehlerbehandlung	_{NOEDT/EDT/EDTF/EDTP/EDTU}
GEND Datenbeschreibung generieren	
GENM Masken generieren	
GET ISAM-Datei lesen	_Dateiname
HALT Programm beenden	
IDB Generieren aus SAM- oder ISAM Datei	_Dateiname[,PRE=nn][,GRP=XXXX][,BEG=nn]
LDB Generieren aus LEASY-Katalog	_LEASYName[,CISName]
LEA LEASY-Katalog öffnen	_Name
LINF Inhaltsverzeichnis aus LEASY-Katalog ausgeben	[_{Name/Teilname*}]
LSTD Datenbeschreibung auflisten	[_{Name/Teilname*}]
LSTE Fehlertexte auflisten	
LSTI Infotexte auflisten	
LSTM Masken auflisten	[_{Name/Teilname*}]
NOP Nulloperation	
PROC Meldungen unterdrücken	
PUTD Datenbeschreibungen unbedingt generieren	
PUTM Masken unbedingt generieren	
READ SAM-Datei lesen	_Dateiname
RECF RECFORM für LEASY-Datei	_{F/V}
RESET EDT-Speicher löschen	

Kommando Erläuterung	Operand
RGEND Datenbeschreibung regenerieren	[_{Name/Teilname*/*}]
RGENM Masken regenerieren	[_{Name/Teilname*/*}]
SAVE ISAM-Datei schreiben	_Dateiname
SDB Generieren aus LMS-Datei	_Name[,PRE=nn][,GRP=XXXX][,BEG=nn]
SEG D- und E-Segment generieren	_Adr,L[,Feldbed],Name
SINF Inhaltsverzeichnis aus LMS Datei ausgeben	[_{Name/Teilname*}]
SORT Datenbeschreibungen nach Adressen sortieren	
SRC LMS-Datei öffnen	_Dateiname
SYSDTA EDT-Eingabe auf SYSDTA setzen	
TERMINAL EDT-Eingabe auf Terminal	
USETB Lesezeiger auf Anfang setzen	
UREAD Satz von CISGEN lesen	
UWRITE Satz an CISGEN übergeben	
WHY Warum-Funktion	_Codierte Meldung
WRITE SAM-Datei schreiben	_Dateiname

```
*[_Kommentar]
```

Mit diesem Kommando wird ein Kommentar (1 bis 255 Zeichen) erzeugt, der von CISGEN nicht ausgewertet wird.

```
ADB_pppppp
```

A-Segment generieren.

Definition des A-Segments in der Satz-, Transformations- oder Bildbeschreibung.

Es wird der EDT-Speicher gelöscht und folgende drei Sätze aufgebaut.

```
AppppppDSS=J,DSA=J,DSU=N,EOC=N,LOGADR=N,TYP=C,MAXDES=000,
A       HD=HD.pppppp,
A       VD=VD.pppppp
```

D-Segmente werden mit den Funktionen IDB
CDB
SDB angefügt.

```
CDB_[Cn,]Name[,PRE=nn][,GRP=xxxx][,BEG=nn]
```

Generierung von D- und E-Segmenten aus einer COBLUR-Datei.

Cn	Teil der COBLUR-Datei (C1, C2, C3, C4) in dem das Element steht. Standardwert: C2
Name	Name des Elements, aus dem eine Satzbeschreibung generiert wird.
PRE=nn	Präfix-Länge der Namen in der COBLUR-Datei (1- oder 2-stellig). Die ersten n Zeichen der Feldnamen werden nicht in das E-Segment übernommen. Standardwert: PRE=00, d.h. die Namen werden ganz übernommen.
GRP=xxxx	Angabe des 4 stelligen Abschnittnamens (MV-Format). Entfällt diese Angabe, hat der Satz V-Format.
BEG=nn	Angabe der Anfangsadresse (1- oder 2-stellig). Standardwert: BEG=05 für V-Format BEG=09 für MV-Format

Diese Anweisung bezieht sich auf die aktuelle COBLUR-Datei, die mit der COB-Anweisung (vgl. Seite 25) eröffnet wurde.

Es werden D- und E-Segmente generiert und in den EDT-Speicher geschrieben. Dieser Speicher wird fortgeschrieben (zum Rücksetzen/Löschen, siehe RESET-Kommando Seite 66).

Das MV-Format kann generiert werden. Es wird pro Abschnitt ein COBOL-COPY-Element benötigt.

Die D- und E-Segmente stehen im EDT-Speicher-0, die COBOL-Statements im EDT-Speicher-1 (vgl. auch Seite 48: Bearbeiten eines COBOL-COPY-Elements).

```
CINF[_[Cn,]{Name/Teilname*}]
```

Ausgabe des Inhaltsverzeichnisses einer COBLUR-Datei auf SYSOUT.

`Name` Name eines Elements in der COBLUR-Datei, ab dem das Inhaltsverzeichnis ausgegeben wird.

`Teilname*` Teil eines Namens (mindestens 1 Byte). Ab diesem Teilnamen, ergänzt mit Spaces, wird das Inhaltsverzeichnis ausgegeben.

`Cn` C1, C2, C3 oder C4

Nummer des Teils der COBLUR-Datei, in dem der Name oder Teilname gesucht wird. Standardwert: C2

Ausgabe:

1. Zeile:	`FILE:`	`Dateiname`	
2. Zeile:	`NAME`	`PARTITION`	
3.-20. Zeile:	`Name des Elements`	`Partition` (Teil)	C1: Maschinenteil (Environment Division) C2: Datenteil (Data Division) C3: Verarbeitungsteil (Procedure Division) C4: Primärprogramm
21. Zeile:	`WEITER? (J/N)`		J: Auflistung der nächsten 18 Elemente N: Beenden dieser Funktion

Beispiel:

```
COB PERS.COBLIB
CINF PE*

FILE:     PERS.COBLIB
NAME      PARTITION
PERSSATZ  C2
PKOPF     C2
```

```
COB_Dateiname
```

Eröffnen einer COBLUR-Datei.

`Dateiname` Name der COBLUR-Datei (DVS-Name)

Ist schon eine COBLUR-Datei eröffnet, so wird diese zuerst geschlossen. Es kann immer nur eine COBLUR-Datei offen sein.

Die Datei wird nur zum Lesen eröffnet.

Es wird geprüft, ob KEYPOS=5 und KEYLEN=14 erfüllt sind.

```
DEFD
```

Generieren und ergänzen von Satzbeschreibungen

Das folgende Funktionsmenü steuert die Funktionen generieren und ergänzen von Satzbeschreibungen.

```
*** FUNKTIONEN BEARBEITEN SATZBESCHREIBUNG (DEFD) ***

    BITTE NAMEN DER SATZBESCHREIBUNG EINGEBEN
    UND FUNKTION AUSWAEHLEN (ANKREUZEN)

  NAME DER SATZBESCHREIBUNG:    ------

  GENERIEREN SATZBESCHREIBUNG :
  ERGAENZEN  SATZBESCHREIBUNG :


  ENDE                        :
```

Der Satzbeschreibungsname und eine Funktion muß eingegeben werden. Der Name wird auf korrekte Eingabe geprüft und ob unter diesem Namen in der DABEL bereits eine Beschreibung existiert. Die Funktionsauswahl wird auf Eindeutigkeit geprüft.

Generieren einer Satzbeschreibung:

Die Parameter des A-Segments und die Daten der D/E-Segmente werden in getrennten Masken erfaßt.

A-Segment

Die Parameter des A-Segments werden in zwei verschiedenen Masken erfaßt, die folgende Parametergruppen enthalten:

Parameter A-Segment 1. Teil:	Datenschutz Steuerung Dateiangaben
Parameter A-Segment 2. Teil	Druckformat/Bild FILE-Parameter Hauptdatei Spezielle Parameter für NK - ISAM

A-Segment 1. Teil

```
*** PARAMETER A-SEGMENT (1.TEIL) FUER SATZBESCHREIBUNG: SBESCH ***

  DATENSCHUTZ (DSA=,DSS=,DSU=)
   AUSGABE ERLAUBT ?  J  (J/N)
   SUCHEN  ERLAUBT ?  J  (J/N)
   UPDATE  ERLAUBT ?  N  (J/N)
  STEUERUNG (LOGADR=,MAXDES=,EK=,TK=,SQLTAB=)
   ORDNUNGSBEGRIFF VON CIS ?     J       (J/N)
   MAX.LAENGE BEI INVERTIERUNG     0     (0<=LAENGE<=238-LNG(OB))
   TEXTTRENNER VOLL-   TEIL-             (MAX.12 ZEICHEN, NICHT "E";
               INVERTIERUNG              SPACE NUR,WENN # SPACE FOLGT!)
   LINEARISIERTE MULTIPLITAETEN? N  (N/J)
  DATEIANGABEN (EOC=,HD=,HDLINK=,VD=,VDLINK=)
   EXPLIZIT. OPEN HD-/VD-DATEI   N   (N/J)
   NAME HD-DATEI (MIT/OHNE CAT-ID/USER-ID)
                 HD.SBESCH
     ODER HD-LINKNAME
   NAME VD-DATEI (MIT/OHNE CAT-ID/USER-ID)
                 VD.SBESCH
     ODER VD-LINKNAME
  WEITERE A-SEGMENT-PARAMETER ? N   (J=JA:   AUSGABE-/KATAL.-PARAM.)
                                    (N=NEIN: ERFASSEN D-SEGMENTE)
```

Die Mehrzahl der möglichen Eingaben ist mit Standardwerten vorbelegt. In der Überschriftszeile und in die Felder NAME HD-DATEI und NAME VD-DATEI wird der eingegebene Beschreibungsname aus der Funktionsmaske eingesetzt.

Alle Werte können überschrieben werden. Die Eingabemöglichkeiten zu den einzelnen Feldern sind jeweils in Klammern dargestellt. Bei den Eingabegruppen ist in Klammern angegeben, für welche A-Segment-Parameter die folgenden Felder gelten.

Wenn keine Invertierung gewünscht wird (d.h. wenn der Wert für MAXDES auf "0" bleibt), muß der vorbelegte VD-Dateiname gelöscht werden.

Die Eingabedaten werden auf Richtigkeit und Plausibilität geprüft. Bei Vorliegen eines Fehlers wird die Maske mit einer entsprechenden Fehlermeldung ausgegeben.

Die Steuerungsparameter LOGADR, DSA und DSS enthalten im Gegensatz zur Erfassung mit dem EDT den Standardwert "J".

Die sonstige Bedeutung der einzelnen Felder entspricht der Beschreibung ab Seite 85.

Wenn der zweite Teil der A-Segment-Parameter erfaßt werden soll, muß in der letzten Textzeile J eingegeben werden.

A-Segment 2. Teil

In die Erfassungsmaske für den zweiten Teil der A-Parameter wird nur verzweigt, wenn sie aus der Maske für den ersten Teil angewählt wurde.

```
*** PARAMETER A-SEGMENT (2.TEIL) FUER SATZBESCHREIBUNG: SBESCH ***

  DRUCKFORMAT/-BILD (KOPF=,ZEIT=,FORM=,ZEILEN=)
   UEBERSCHRIFTSTEXT (MAX.60)  :

   UEBERSCHRIFT MIT UHRZEIT ?  :  N      (N/J, IN DER FORM HH.MM)
   ZEILENANZAHL TITELSEITE     :  72     (72 IST MAXIMUM)
   ZEILEN PRO SEITE/LEERZEILEN :  50 / 0 (MAXIMUM: 72/0;Z.B. 36/1)
  FILE-PARAMETER FUER HD-DATEI (FCBTYPE=,BLKSIZE=,RECFORM=,RECSIZE=,SPACE=)
   DATEIFORMAT                 :  ISAM   (ISAM/SAM)
   BLOCKUNGSFAKTOR (STD,N)     :   1     (1 <= N <= 16)
   SATZFORMAT                  :  V      (V/F VARIABLE/FESTE SATZLAENGE)
   SATZLAENGE                  :      0  (MUSS BEI SATZFORMAT=F ANGEG.SEIN)
   DATEIGROESSE (PRIMAER/SEK.) :      3,    3   (STD: 3*BLOCK.-FAKTOR)
  SPEZIELLE PARAMETER FUER NK-ISAM (POOLLNK,BLKCTRL)
   HDPOOLLNK                      HDBLKCTRL        (DATA/NO)
   VDPOOLLNK                      VDBLKCTRL        (DATA/NO)

  ERFASSEN D-SEGMENTE ?  J  (J=JA, N=NEIN: ENDE)
                            (A=ABBRUCH, A-SEGMENT WIRD N I C H T ERZEUGT !)
```

Alle Eingabefelder sind mit Standardwerten vorbelegt. In der Überschriftszeile wird der eingegebene Beschreibungsname aus der Funktionsmaske eingesetzt.

Alle Werte können überschrieben werden. Die Eingabemöglichkeiten zu den einzelnen Feldern sind jeweils in Klammern dargestellt. Bei den Eingabegruppen ist in Klammern angegeben, für welche A-Segment-Parameter die folgenden Felder gelten.

Die Eingabedaten werden auf Richtigkeit und Plausibilität geprüft. Bei Vorliegen eines Fehlers wird die Maske mit einer entsprechenden Fehlermeldung erneut ausgegeben.

In dieser Maske gibt es drei Möglichkeiten der Fortsetzung:

- J: Es findet anschließend die Eingabe für die D-/E-Segmente statt (Standardwert).

- N: Die Erfassung wird unterbrochen. Die D-/E-Segmente können später erfaßt werden.

- A: Abbruch. Alle bereits eingegebenen Parameter des A-Segments werden n i c h t gespeichert. Es wird keine Satzbeschreibung in der DABEL angelegt.

Der Ausgabeparameter KOPF (Überschriftstext) kann statt 30 Zeichen bei der Erfassung mit dem EDT, maximal 60 Zeichen enthalten.

Die sonstige Bedeutung der einzelnen Felder entspricht der Beschreibung ab Seite 90.

D-/E-Segment

```
*** EINGABE D/E-SEGMENT FUER SATZBESCHREIBUNG: SBESCH ***


  ASBESCHDSA=J,DSS=J,DSU=N,LOGADR=J,MAXDES=20,EOC=N,
  A      HD=HD.SBESCH,
  A      VD=VD.SBESCH
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+
   FELDBEZEICHNUNG                  (MAX: 15 BUCHST. UND/ODER ZIFFERN)
    ABSCHNITTSNAME           AB ADRESSE        5    IN LAENGE       0
   LOGISCHE BEDEUTUNG             :     ART DER DARSTELLUNG           :
     (ORDNUNGSBEGRIFF        = O)        (BINAER                 = B)
     (COMPOUND-KEY           = Z)        (GEPACKT                = P)
     (UNTER-/UNTERTEILT.FELD = U)        (ENTPACKT               = R)
     (PSEUDO-/SORTPSEUDOFELD = P/S)      (ZEICHEN                = T)
     (VARIABLE FELDFLAENGE   = V)        (GEPACKT OHNE VORZEICH  = H)
     (KOMPRIMIERTES FELD     = K)        (ZEICHEN MIT END-NULLEN = L)
     (WIEDERHOLFELD          = W)        (ZEICHEN M.FUEHR.SPACES = A)
   WIEDERHOLABSCHNITT (N/J)      : N   KOMMASTELLEN (NUR ZU P/R/H)   :  0
  UEBERNAHME IN VERWEISDATEI (N/J) N   KURZ-BEZ.
   ART:  STICHWORT?    OHNE NULLWERT?    DUPKEYS?    GROßBUCHST.?
  EINGABE WEITERES D-SEGMENT? J   (J=JA; N=NEIN, ENDE)
```

In der Überschriftszeile wird der Satzbeschreibungsname aus der Funktionsmaske übernommen.

In den sechs auf die Überschriftszeile folgenden Zeilen werden immer die letzten Eingaben im EDT-Format dargestellt.

Einige Eingabefelder sind mit Standardwerten vorbelegt, die jedoch bei Bedarf geändert werden können.

Im vorstehenden Beispiel ist der Wert für MAXDES verändert worden. Die Eingaben für den zweiten Teil des A-Segments wurden entweder übersprungen oder die Standardwerte wurden nicht verändert und deshalb auch nicht explizit übernommen.

Wo es sinnvoll erscheint, wird der Abschnittsname vorbelegt und eine errechnete Adresse vorgegeben. Beide Werte können überschrieben werden, wenn sie für die aktuelle Eingabe nicht zutreffen.

Die Eingabe für das Feld KOMMASTELLEN kann zwischen 1 und 15 variieren und ist nur bei den DARSTELLUNGSARTEN "P", "R" oder "H" möglich.

Wenn eine Übernahme in die VD stattfinden soll, kann eine besondere Art der Übernahme gewählt werden. Die in der Maske angebotenen Möglichkeiten schließen sich gegenseitig aus, nur die Kombination Stichwort/Großbuchstaben kann gleichzeitig angekreuzt werden. Die Stichwort-invertierung ist nur bei der Darstellungsart "T" möglich, nicht jedoch beim Compound-Key Oberbegriff.

Die sonstige Bedeutung der einzelnen Felder entspricht der Beschreibung ab Seite 94.

Die Einhaltung der Eingaberegeln, wie ab Seite 94 beschrieben, wird nach Absenden mit "DUE" überprüft. Außerdem werden noch weitere Plausibilitäten zum A-Segment durchgeführt. Im Fehlerfall wird die Maske mit einer entsprechenden Fehlermeldung erneut ausgegeben.

Nach korrekter Eingabe werden die eingegebenen Informationen in ein D-/E-Segment im EDT-Format umgewandelt. Wenn ein weiteres D-Segment eingegeben wird (Vorbelegung !), wird die Eingabemaske erneut ausgegeben. Das eben eingegebene Segment wird im Anzeigebereich nach der Überschriftszeile ausgegeben.

Beendigung der Erfassung:

Die Erfassung wird beendet indem bei "EINGABE WEITERES D-SEGMENT" N eingegeben wird. Sollten beim Verlassen der Erfassung noch Daten in der Maske stehen, so werden diese n i c h t übernommen!

Aus den erfaßten Eingabezeilen, die rollierend im Anzeigeteil dargestellt wurden, wird die Satzbeschreibung im DABEL-Format erzeugt. Anschließend wird die Beschreibung wie beim Kommando GEND auf SYSLST protokolliert.

Ergänzen einer Satzbeschreibung:

Vorhandene Satzbeschreibungen, die im Dialog mit DEFD erfaßt wurden, können um weitere D-Segmente erweitert werden.

Es erfolgt die Ausgabe der Maske für die Erfassung der D-/E-Segmente, wobei die letzten sechs Zeilen der ausgewählten Satzbeschreibung im Anzeigeteil nach der Überschriftszeile dargestellt werden.

Es erfolgen Prüfungen auf eindeutige Funktionsauswahl, auf korrekte Eingabe des Beschreibungsnamens und darauf, ob unter diesem Namen in der DABEL eine Beschreibung vorhanden ist. Außerdem wird geprüft ob es sich um eine Satzbeschreibung handelt und ob sie im Dialog, also mit dem Kommando DEFD, erfaßt wurde.

Die Erfassung der D-Segmente erfolgt wie auf Seite 29 beschrieben.

DEFM

Definition der Masken.

Menügesteuerte Generierung und Änderung der Bildschirmmasken und Folgemasken.

Der Aufbau von Masken ist im Kapitel - Bildschirmmasken - (Seite 125) beschrieben.

Beispiel:

```
*DEFM              (Ausgabe des Funktionsmenüs am Bildschirm)

*** FUNKTIONSMENUE ***

    BITTE NUR EIN FELD ANKREUZEN UND NAMEN
    DER BILDSCHIRMMASKE EINGEBEN

    GENERIEREN BILDSCHIRMMASKE : X
    AENDERN BILDSCHIRMMASKE    :
    ENDE                       :

    NAME BILDSCHIRMMASKE       : MASK01



*** GENERIEREN BILDSCHIRMMASKE ***

    BITTE NAMEN DER SATZBESCHREIBUNG UND EVTL.
    NAMEN DES PLAUSIBILITAETENPROGRAMMS EINGEBEN

    NAME BILDSCHIRMMASKE          :  MASK01
    NAME SATZBESCHREIBUNG         :  DATEN1
    NAME PLAUSIBILITAETENPROGRAMM :  ------
    ABBRUCH                       :

DB71    LSP-TASTE DRUECKEN, DANN MASKE EINGEBEN
```

(Nach der Maskeneingabe (vgl. Seite 129) wird die Generierung einer Folgemaske angeboten.)

```
*** FOLGEMASKE ***

    BITTE NAMEN DER FOLGEMASKE
    EINGEBEN/MODIFIZIEREN

    NAME FOLGEMASKE: ------
```

Wird kein Name angegeben, erscheint das Funktionsmenü. Die Generierung der Bildschirmmaske kann durch Ankreuzen von 'ENDE' beendet werden.

```
DEFT
```

Generieren von Transformationsbeschreibungen

Anhand einer Quell-Satzbeschreibung und einer Ziel-Satzbeschreibung wird eine Transformationsbeschreibung erstellt.

```
*** FUNKTION ERZEUGEN TRANSFORMATIONSBESCHREIBUNG (DEFT) ***

    BITTE NAMEN DER BESCHREIBUNGEN EINGEBEN
      ODER ENDE-FELD AUSWAEHLEN

  NAME DER TRANSFORMATIONSBESCHREIBUNG:  ------

     ERZEUGEN   AUS QUELL-BESCHREIBUNG:  ------
                FUER ZIEL-BESCHREIBUNG:  ------


  ENDE                        :
```

Quell- und Zielbeschreibung müssen als Satzbeschreibungen in der DABEL vorhanden sein. Die angegebene Transformationsbeschreibung darf noch nicht existieren. Die Existenz von D-Segmenten in der Quell- und Zielbeschreibung wird geprüft.

Im Fehlerfall wird eine entsprechende Fehlermeldung ausgegeben.

1. Feldname und die dazugehörende Feldlänge der Quell- und der Zielbeschreibung sind identisch:

   Die Transformationsbeschreibung wird ohne Ausgabe einer weiteren Maske generiert und in den EDT-Bereich gestellt, um sie überprüfen zu können.

2. Feldname der Zielbeschreibung ist nicht in der Quellbeschreibung vorhanden:

Folgende Maske wird ausgegeben:

```
*** FEHLENDES EINGABEFELD ZUR TRANSFORM.-BESCHREIB.: TRANSB ***

    ZUM AUSGABEFELD  ZIELFELD3
       IN ABSCHNITT         AB POSITION  0016   IN LAENGE  008
       DARSTELLUNGSFORM  T
       FEHLT EIN KORRESPONDIERENDES EINGABEFELD

    SOLL DAS AUSGABEFELD   DAS TAGESDATUM AUFNEHMEN ?
                           (IN DER FORM TT.MM.JJJJ)

                 O D E R   STANDARDFÜELLUNG ERHALTEN ? X
                           (NULLEN BZW. BLANKS)

                 O D E R   AUS EINER KONSTANTEN
                           MIT FOLGENDEM INHALT



                           GEFUELLT WERDEN ?
```

ZIELFELD3 Fehlendes Feld in der Quellbeschreibung mit Angabe des Abschnittsnamens (sofern vorhanden), der Position, Länge, und Darstellungsform in der Zielbeschreibung.

Auswahlmöglichkeiten:

- Standardfüllung: Wenn das Zielfeld eine Standardfüllung erhalten soll, so muß die Maske unverändert abgeschickt werden, da die Auswahl "Standardfüllung" bereits vorbelegt ist.

  Es wird eine Füllkonstante (F-Segment) erzeugt, die in Abhängigkeit von der Darstellungsform des Zielfeldes aus Blanks, Nullen oder low-values in der Länge des Zielfeldes besteht.

- Tagesdatum: Die Auswahl von "Tagesdatum" ist nur sinnvoll, wenn das Zielfeld ein 10 Byte langes Textfeld ist.

- Konstante: Nur bei der Darstellungsform T (Textfeld) oder R (Ziffern rechtsbündig) ist die Auswahl "Konstante" möglich. Es muß eine der Darstellungsform entsprechende Konstante in Länge des Zielfeldes eingegeben werden. Es wird mit dieser Konstanten ein F-Segment erzeugt.

  Die maximale Eingabelänge beträgt 80 Stellen. Wenn eine Konstante mit Blanks endet, muß hinter der benötigten Länge ein abdruckbares Zeichen (z.B. "*") eingegeben werden, da sonst die Eingabelänge infolge der Vorbelegung des Eingabefeldes nicht richtig erkannt wird.

3. Feldname ist in der Quellbeschreibung vorhanden, die dazugehörende Feldlänge ist kleiner als die des entsprechenden Feldes in der Zielbeschreibung.

   Folgende Maske wird ausgegeben:

```
*** KUERZERES EINGABEFELD ZUR TRANSFORM.:-BESCHREIB.: TRANSB ***

    ZUM AUSGABEFELD  ZIELFELD5
       IN ABSCHNITT         AB POSITION  0029   IN LAENGE  004
       DARSTELLUNGSFORM  R   DIFF.-LAENGE  001
       GIBT ES NUR EIN KUERZERES EINGABEFELD

    SOLL DIE RESTLAENGE STANDARDFUELLUNG ERHALTEN ?    X
                 (FUEHRENDE NULLEN ODER END-BLANKS)

                 O D E R   AUS EINER KONSTANTEN
                           MIT FOLGENDEM INHALT


                           GEFUELLT WERDEN ?
```

   Für diese Maske gelten die bei fehlendem Quellfeld gemachten Aussagen in gleicher Weise. Als Feldlänge des F-Segments ist hier jedoch nur die Differenzlänge zwischen Ziel- und Quellfeld zu verstehen.

   Im dargestellten Beispiel war die Darstellungsform von Quell- und Zielfeld die gleiche. Wenn unter dieser Bedingung der Umkehrfall, also ein kürzeres Zielfeld, auftritt, wird der Feldinhalt des Quellfeldes für die Transformation in Abhängigkeit vom Format rechts oder links entsprechend verkürzt.

   Bei unterschiedlicher Darstellungsform des Quell- und Zielfeldes sind von den bei CISGEN vorgesehenen Kombinationen nur die Umwandlungen von/nach R,P,H und B erlaubt. Die Längen von Quell- und Zielfeld müssen zur jeweiligen Kombination passen:

   Z.B.: von R 5-stellig nach P 3-stellig
         von H 2-stellig nach R 4-stellig

```
DELD_{Name/Teilname*/*}
```

Löschen der Datenbeschreibungen in der zugewiesenen DABEL.

`Name`	Name der zu löschenden Datenbeschreibung.
`Teilname*`	Alle Datenbeschreibungen, deren Namen mit "Teilname" anfangen, werden gelöscht.
`*`	Alle Datenbeschreibungen werden gelöscht.

Die Anzahl der gelöschten Datenbeschreibungen wird protokolliert.

```
DELM_{Name/Teilname*/*}
```

Löschen der Masken in der zugewiesenen Datenbeschreibungsdatei.

`Name`	Name der zu löschenden Maske.
`Teilname*`	Alle Masken, deren Namen mit "Teilname" anfangen, werden gelöscht.
*	Alle Masken werden gelöscht.

Die Anzahl der gelöschten Masken wird protokolliert.

```
DINF_{Name/Teilname*}
```

Ausgabe des Inhaltsverzeichnisses der DABEL auf SYSOUT.

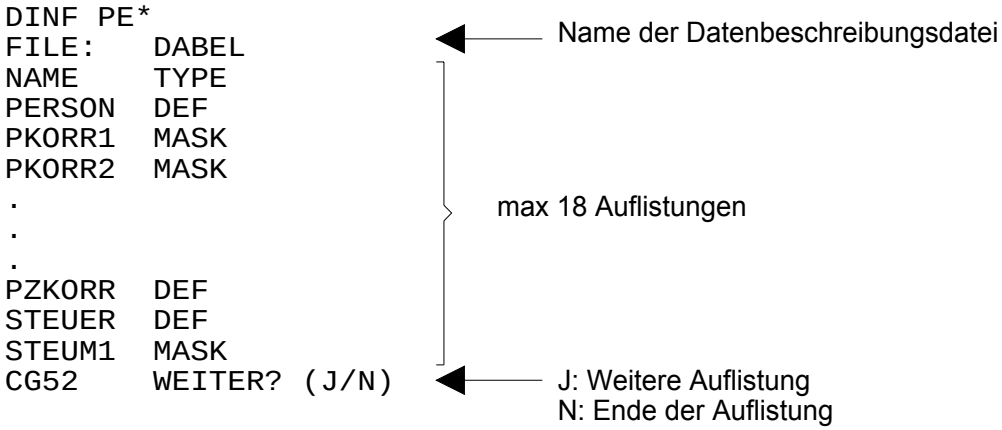
`Name` Name eines Elements in der DABEL, ab dem das Inhaltsverzeichnis auszugeben ist.

`Teilname*` Teil eines Namens (mindestens 1 Byte). Ab diesem Teilnamen, ergänzt mit Spaces, wird das Inhaltsverzeichnis ausgegeben.

Datenbeschreibungen und Bildschirmmasken werden alphabetisch aufgelistet. Die Typ-Angabe 'DEF' (vgl. Beispiel) charakterisiert eine Datenbeschreibung (Satz-, Transformations- oder Bildbeschreibung), die Typ-Angabe 'MASK' eine Bildschirmmaske.

Sollen nur Datenbeschreibungen aufgelistet werden, so ist die DINFD-Anweisung (vgl. Seite 38) zu verwenden. Mit der DINFM-Anweisung (vgl. Seite 39) werden nur die Bildschirmmasken aufgelistet.

Beispiel:



```
DINFD_{Name/Teilname*}
```

Ausgabe des Inhaltsverzeichnisses der in der DABEL vorhandenen Datenbeschreibungen auf SYSOUT.

Name	Name des Elements in der DABEL, ab dem das Inhaltsverzeichnis auszugeben ist.
Teilname*	Teil eines Namens (mindestens 1 Byte), ab dem, ergänzt mit Spaces, das Inhaltsverzeichnis ausgegeben wird.

Diese Anweisung arbeitet analog der DINF-Anweisung (vgl. Seite 37). Zusätzlich wird, falls vorhanden, der HD-Parameter aus dem A-Segment der Datenbeschreibung angegeben.

Beispiel:

```
DINFD
FILE:   DABEL
NAME    TYPE  PARAMETER
ABSBEZ  DEF   HD=HD.ABSBEZ
ABSD01  DEF   HD=HD.ABSBEZ
ADILOS  DEF   HD=HD.ADILOS
ADITRA  DEF   HD=HD.FMULTI
ADRAUF  DEF   HD=                              (kein HD-Parameter im A-Segment)
ADR001  DEF   HD=$CISKURS.HD.KUNDEN
ADVO02  DEF   HD=HD.MESSE1
AIPAMT  DEF   HD=HD.AIPAMT
AKTUEL  DEF   HD=HD.DBANK
```

```
DINFM_{Name/Teilname*}
```

Ausgabe des Inhaltsverzeichnisses der in der DABEL vorhandenen Masken auf SYSOUT.

Name	Name des Elements in der DABEL, ab dem das Inhaltsverzeichnis auszugeben ist.
Teilname*	Teil eines Namens (mindestens 1 Byte), ab dem, ergänzt mit Spaces, das Inhaltsverzeichnis ausgegeben wird.

Diese Anweisung arbeitet analog der DINF-Anweisung (vgl. Seite 37). Es wird, falls vorhanden, der HD-Parameter aus dem A-Segment der Datenbeschreibung angegeben.

Beispiel:

```
DINFM
FILE:   DABEL
NAME    TYPE  PARAMETER
ABSMA1  MASK  DEF=ABSBEZ
ABSM01  MASK  DEF=ABSD01
ADIFOR  MASK  DEF=ADILOS
ADRMA1  MASK  DEF=ADR001
AENTEI  MASK  DEF=SOMMER
AF12TE  MASK  DEF=FMULTI
AKTUEF  MASK  DEF=AKTUEL
.
.
.
AMMASK  MASK  DEF=AMTEST
AMMAS1  MASK  DEF=AMTEST
CG52    WEITER? (J/N)
```

```
DSWITCH_Dateiname
```

Umschalten auf eine andere DABEL.

`Dateiname` Name der Datenbeschreibungsdatei (DABEL)

Die aktuelle DABEL wird geschlossen und die im DSWITCH-Kommando angegebene DABEL geöffnet. Es kann immer nur eine DABEL offen sein.

Die Datei wird im Änderungsmodus eröffnet.

Ist die Datei noch nicht vorhanden, so wird sie eingerichtet.

```
Format 1:      EDT
Format 2:      EDTP
Format 3:      EDTU
Format 4:      EDTF

Format 5:      EDT: @Kommando
```

EDT aufrufen - Wahl des EDT-Modus.

Der Inhalt des EDT-Speichers kann mit EDT-Kommandos bearbeitet werden.

CISGEN als Dialogprogramm:

EDT	ruft	EDT-L auf
EDTP	ruft	EDT-L auf mit @Print
EDTU	ruft	EDT-L auf mit @UPDATE
EDTF	ruft	EDT-F auf

Rücksprung ins Dialogprogramm CISGEN:

bei: EDT, EDTP und EDTU mit @RETURN

bei: EDTF mit HALT

CISGEN als Unterprogramm:

EDT: ist ein CISGEN-Passiv-Kommando, mit dem jedes zulässige EDT-L-Kommando '@Kommando' abgesetzt werden kann. Anschließend verlangt CISGEN eine neue Eingabe.

Soll beispielsweise eine Zeile in den EDT-Speicher geschrieben werden, so ist vom aufrufenden Programm an CISGEN das Kommando

```
EDT:@CREATE 300:'...'
```

zu übergeben.

```
ERRPUT_{NOEDT/EDT/EDTF/EDTP/EDTU}
```

Fehlerbehandlung beim Generieren.

Standardwert: NOEDT

Die Funktionen (nur jeweils ein Element kann sich im EDT-Speicher befinden)

GEND
GENM
PUTD
PUTM
SORT beinhalten folgende Fehlerbehandlung:

In Abhängigkeit des Operanden erfolgt die Fehlerbehandlung wie folgt.

Operand	Verarbeitungsschritte
fehlt NOEDT	Ausgabe der Fehlermeldung
EDT EDTP EDTU EDTF	1. Ausgabe der Fehlermeldung 2. Verzweigung in den EDT mit dem Modus = Operand vgl. vorhergehende Seite 3. Nach Rücksprung aus dem EDT, Wiederholung der 1. Funktion

```
GEND
```

Datenbeschreibungen generieren (bedingt).

Alle im EDT-Speicher (Arbeitsdatei-0) stehenden Datenbeschreibungen werden in die gerade offene DABEL geschrieben. Ist eine Datenbeschreibung mit gleichem Paßwortes schon in der DABEL vorhanden, muß folgende Meldung beantwortet werden.

```
DB93     VORHANDENE DB UEBERSCHREIBEN?
         (J/N)    Name der Datenbeschreibung
```

J überschreiben
N nicht überschreiben

IM EDT-Speicher vorhandene Masken werden ignoriert.

Jede Datenbeschreibung wird auf SYSLST protokolliert.

```
GENM
```

Masken generieren (bedingt).

Alle im EDT-Speicher (Arbeitsdatei-0) stehenden Masken werden in die gerade offene DABEL geschrieben. Ist eine Maske mit gleichem Paßwortes schon in der DABEL vorhanden, muß folgende Meldung beantwortet werden.

```
DB93     VORHANDENE DB UEBERSCHREIBEN?
         (J/N)    Name der Datenbeschreibung
```

J überschreiben
N nicht überschreiben

IM EDT-Speicher vorhandene Datenbeschreibungen werden ignoriert.

Jede Maske wird auf SYSLST protokolliert.

```
GET_Dateiname
```

Lesen einer ISAM-Datei.

`Dateiname` Name der Datei, die gelesen werden soll (die GET-Anweisung wird an den EDT weitergereicht und in @GET'Dateiname' umgeformt).

Die gelesenen Sätze werden in den EDT-Speicher (Arbeitsdatei-0) geschrieben. Der Speicher wird fortgeschrieben.

Soll ab Anfang des Speichers geschrieben werden, so muß vorher die RESET-Anweisung (vgl. Seite 66) angewandt werden.

Die Anzahl der im EDT-Speicher stehenden Datenbeschreibungen und Masken wird angezeigt.

Ebenso können mit der GET-Anweisung SAM-Dateien gelesen werden. Der EDT gibt jedoch die Meldung

```
SAM FILE - DOING @READ
```

aus.

```
HALT
```

Beenden des Programms CISGEN.

Alle offenen Dateien werden geschlossen.

Wird CISGEN als Unterprogramm benützt, so übergibt CISGEN die Meldung, IM90, an das Hauptprogramm.

```
IDB_Dateiname[,PRE=nn][,GRP=xxxx][,BEG=nn]
```

SAM-/ISAM-Datei: D- und E-Segmente generieren. Generierung von D- und E-Segmenten aus einem COBOL-COPY-Element, das sich in einer SAM- oder ISAM-Datei befindet.

`Dateiname`	Name der Datei, die ein COBOL-COPY-Element enthält.	
`PRE=nn`	Präfix-Länge der Namen in der Datei (1- oder 2-stellig). Dieses Präfix wird nicht als Name in das E-Segment übernommen. Standardwert: PRE=00	
`GRP=xxxx`	Angabe eines Abschnittnamens (MV-Format). Fehlt der Operand GRP werden Sätze mit V-Format beschrieben.	
`BEG=nn`	Angabe einer Anfangsadresse (1- oder 2-stellig):	BEG=05 für V-Format BEG=09 für MV-Format

Die angegebene Datei kann eine SAM-Datei oder eine ISAM-Datei mit KEYPOS=5 und KEYLEN=8 sein.

Aus dem COBOL-COPY-Element werden D- und E-Segmente generiert und in den EDT-Speicher (Arbeitsdatei 0) geschrieben. Dieser Speicher wird fortgeschrieben. Zum Rücksetzen/Löschen muß das RESET-Kommando (vgl. Seite 66) angewandt werden.

MV-Formate können generiert werden. Pro Abschnitt wird ein COPY-Element benötigt.

Die D- und E-Segmente stehen im EDT-Speicher 0, die COBOL-Statements im EDT-Speicher 1.

Vgl. auch Seite 48: Bearbeiten eines COBOL-COPY-Elements.

Bearbeiten eines COBOL-COPY-Elements

Ein COBOL-COPY-Element kann aus einer

- SAM-/ISAM-Datei (mit der IDB-Anweisung)
- LMS-Bibliothek (mit der SDB-Anweisung)
- COBLUR-Bibliothek (mit der CDB-Anweisung)

in eine CIS-Datenbeschreibung umgesetzt werden (ohne A-Segment).

Die COBOL-Statements werden in komprimierter Form verarbeitet, d.h. pro Statement wird ein Satz gebildet und Spaces auf 1 Space reduziert.

Die COBOL-Statements, die in Spalte 7 kein Space enthalten (z.B. Kommentare) werden ignoriert. Alle anderen Statements, die ein Feld beschreiben (Ausnahme FILLER), werden in ein D- und E-Segment umgewandelt.

Die komprimierten COBOL-Statements werden in die EDT-Arbeitsdatei 1 geschrieben, die generierten D- und E-Segmente in Arbeitsdatei 0.

Jedes COBOL-Statement wird nach COBOL-Worten untersucht. Alle COBOL-Worte, die in der unten aufgeführten Liste nicht auftreten, erzeugen eine Fehlermeldung. Sollte ein COPY-Element komplizierter sein, so muß es manuell bearbeitet werden.

Liste der COBOL-Worte und deren Verarbeitung

FILLER — Das Feld wird verarbeitet, aber es wird kein D- und E-Segment generiert. Die Adresse wird weitergeschaltet. Wenn PRE=nn größer als die Länge eines Feldnamens ist, so wird das Feld als FILLER behandelt.

PIC
PICTURE — gefolgt von:

{ A...A | A(n) | X...X | X(n) } — ergibt Feldbedeutung: T

[S] { 9...9 | 9(n) } [V { 9...9 | 9(n) }] — ergibt Feldbedeutung: R

Beim Format S9...V9... wird die Anzahl der Dezimalstellen ermittelt.

Die Feldbedeutung R wird überschrieben, wenn COMP oder COMP-3 folgt.

COMP
COMPUTATIONAL ergibt Feldbedeutung: B

Die Länge in Bytes wird wie folgt gesetzt:

Stellen in COBOL	Länge in CIS
1 - 4	2
5 - 9	4
> 9	8

COMP-3
COMPUTATIONAL-3 ergibt Feldbedeutung: P

Die Länge in Bytes wird wie folgt berechnet:

$$\frac{(\text{COBOL-Länge}) + 2}{2} \quad \text{(abgerundet)}$$

OCCURS — Kann mehrstufig vorhanden sein (die maximale Schachtelungstiefe ist 10). Bei größerer Schachtelungstiefe Fehlermeldung *4 (Spalte 2 - 3).

Für jedes Feld wird nur ein D- und E-Segment generiert. Die Adressen werden hochgerechnet.

Bei Schachtelungstiefe 1 wird ein Wiederholfeld generiert, bei Schachtelungstiefen größer 2 und kleiner 10 werden die Spalten 2 - 3 mit *3 überschrieben. Sie müssen nachbearbeitet werden.

REDEFINES — Die Adresse wird zurückgeschaltet.
Die interne Logik des Programms wertet nicht die Namen, sondern die Stufennummern aus.

SYNC
SYNCHRONIZED — Wird nicht verarbeitet.

Liefert Fehler *1 und Adressen mit ? ab diesem D-Segment.

VALUE — Wird überlesen, ebenso der Wert.

sonstige
COBOL-Wörter — Fehler *0 und Adressen mit ? ab diesem D-Segment

Fehlermeldungen/Warnungen in Byte 2 und 3 der Datenbeschreibung

*0	Unbekanntes COBOL-Wort	x)
*1	SYNC/SYNCHRONIZED	x)
*2	Fehler bei REDEFINES	x)
*3	OCCURS-Feld mit Schachtelungstiefe ≥ 2	
*4	Fehler bei OCCURS	x)
*5	Fehler bei OCCURS, Increment (Byte 29 - 31) > 255	

x) Ab der in Spalte 2 - 3 eingetragenen Fehlermeldung werden keine Adressen mehr generiert. (? in Byte 11)

Generiertes D-/E-Segment

Byte	Inhalt	
2	Space	oder * bei Fehler
3	D	oder 0/1/2/3/4 bei Fehler
4 - 7	Abschnittsart	falls GRP-Parameter angegeben
8 - 11	Adresse	rechtsbündig ohne führende Nullen
12 - 14	Länge	rechtsbündig ohne führende Nullen Eintrag maximal 255, auch wenn die Länge größer ist
16	Space	oder Wert 1 - 9 falls Dezimalstellenangabe S9...V9...
17	Feldbedeutung	A falls PIC A T falls PIC X R falls PIC S9/9 B falls COMP P falls COMP-3
45	E	
46 - 60	Feldbezeichnung	max. 15 Bytes

Alle nicht beschriebenen Bytes enthalten Spaces.

```
LDB_LEASY-Name[,CIS-Name]
```

Datenbeschreibung aus LEASY-Katalog generieren.

`LEASY-Name`	Name des Elements im LEASY-Katalog, aus dem generiert werden soll.
`CIS-Name`	Name des Elements im CIS-Katalog (6 Bytes = CIS-Paßwort) Standardwert: Die 6 ersten Bytes des LEASY-Namens

Diese Anweisung bezieht sich auf den aktuellen LEASY-Katalog, der mit der LEA-Anweisung (vgl. Seite 54) eröffnet wurde.

Nach Löschung des EDT-Speichers werden A-Segmente generiert. Sind im Element des LEASY-Katalogs Sekundärindizes vorhanden, so wird für jeden von ihnen ein D- und E-Segment generiert.

Voraussetzungen:
- LEASY-TYP = S (Stammdatei)
- FCBTYPE = SAM/ISAM/PAM
- RECFORM = V/F
- RDPASS = N (Wenn J muß das /PASSWORD-Kommando beim Ablauf von CISGEN angegeben werden.)

Das LDB-Kommando versucht mit SHOW-FILE-ATTRIBUTES den Satztyp (RECFORM) der Datei zu eruieren. Ist nur der LEASY-Katalog vorhanden und somit der Satztyp nicht festgestellbar, so kann vor der LDB-Anweisung mit dem RECF-Kommando (vgl. Seite 65) der Satztyp vorgegeben werden, andernfalls wird im Dialog die RECFORM angefordert.

Die Generierung der Datenbeschreibung aus SAM-Dateien ist nicht sinnvoll, da LEASY für SAM-Dateien keine SI-Dateien (Verweisdateien) führt und daher nur ein A-Segment generiert werden würde. SAM-Dateien können mit CIS bearbeitet werden.

RECFORM=V bei PAM-Dateien: Die ersten 4 Bytes des Satzes sind vom Satzlängenfeld belegt.

Generiertes A-Segment:

```
 AppppppDSS=J,DSA=J,DSU=N,EOC=N,LOGADR=N,TYP=L,MAXDES=m
[A      RECFORM=r,RECSIZE=rrrrr,]  (nur bei PAM)
 A      HD=Name[,]
[A      VD=Name-SI]
```

pppppp	Name der CIS-Datenbeschreibung.
TYP=L	Sekundärindizes liegen im LEASY-Format vor.
MAXDES=m	Länge des längsten Sekundärschlüssels.
RECFORM=r	Satztyp bei PAM-Datei: F - Datei enthält Sätze fester Länge. V - Datei enthält Sätze variabler Länge.
RECSIZE=rrrrr	Maximale Satzlänge, falls RECFORM=V, bei PAM-Dateien lt. Eintrag des LEASY-Katalogs.

Generiertes D-Segment:

Für jeden Sekundärindex wird folgendes D-/E-Segment gebildet:

SEGM	QUELL DEFINITION			LOG DARST	VT	ZIEL-DEFINITION			SONDER FUNKTION		WA	SEGM	FELDBEZEICHNUNG TRANSFORMATIONS-ERGAENZUNG
	AA	ADR	LNG			AA	ADR	LNG	SF1	SF2			
3	4	8	12	15	18	21	25	29	32	36	0	5	46 (41-44 FREI)
D	aaaa		lll	f T					iiii			E	nnnnn.....

aaaa — Adresse (rechtsbündig, ohne führende Nullen)

Bei RECFORM=F wird zur Adresse 4 (4 Bytes Satzlängenfeld) addiert.
Bei einem zusammgesetzten Schlüssel ist die Adresse 1.

lll — Länge (rechtsbündig, ohne führende Nullen)

Bei einem zusammengesetzten Schlüssel ist es die Summe aller Längen.

f — Logische Feldbedeutung.

Einfaches Feld	: Space	
Zusammengesetztes Feld	: Z	
Ordnungsbegriff	: O	(Ordnungsbegriff, wenn Adresse und Länge mit KEYPOS und KEYLEN übereinstimmen.)

iii — Kurzname, wird aus der SI-Nummer generiert.

X'01' wird zu C'001'
.
.
.
C'FF' wird zu C'255'

nnnnn — Logischer Name des Sekundärindex.

Die Sätze werden in der LEASY-Reihenfolge generiert. Mit dem Parameter SORT können sie nach aufsteigenden Adressen geordnet werden.

```
LEA_Name
```

Eröffnen eines LEASY-Kataloges.

Name Logischer Katalogname.

(Eigentlicher DVS-Name: Name.LEASYCAT)

Es kann immer nur ein LEASY-Katalog offen sein. Vor Eröffnung eines LEASY-Kataloges wird ein eventuell bereits geöffneter geschlossen.

Die Datei wird nur zum Lesen eröffnet.

Es kann ein LEASY-Katalog der Version 2 oder der Version 3 eröffnet werden.

```
LINF[_{Name/Teilname*}]
```

Inhaltsverzeichnis des LEASY-Katalogs ausgeben.

`Name`	Name eines Elements des LEASY-Katalogs, ab dem das Inhaltsverzeichnis auszugeben ist.
`Teilname*`	Teil eines Namens (mindestens 1 Byte). Ab diesem Teilnamen, ergänzt mit Spaces, wird das Inhaltsverzeichnis ausgegeben.

1. Zeile:	`FILE: Dateiname`
2. Zeile:	`NAME TYP RECFORM RDPASS ANZ.SI (Vn)`
3.-20. Zeile:	Auflistung von 18 Elementen des Inhaltsverzeichnisses
21. Zeile:	`WEITER? (J/N)`

`NAME`	LEASY-Name
`TYP`	LEASY-Dateityp
`RECFORM`	(F oder V) oder Fehlercode von FSTAT
`RDPASS`	J oder N
`ANZ.SI`	Anzahl Sekundärindizes
`Vn`	LEASY-Version

```
LSTD[_{Name/Teilname*}]
```

Datenbeschreibungen auflisten.

`Name`	Name der Datenbeschreibung, ab der aufgelistet werden soll.
`Teilname*`	Teil eines Namens (mindestens 1 Byte). Ab diesem Teilnamen, ergänzt mit Spaces, werden die Datenbeschreibungen aufgelistet.

Ohne Angabe des Namens bzw. Teilnamens, werden alle Datenbeschreibungen aufgelistet.

Pro Datenbeschreibung wird eine Seite auf SYSLST ausgegeben.

Am Ende wird die Anzahl der aufgelisteten Elemente ausgegeben.

LSTE

CIS-Fehlermeldungen ausdrucken.

Alle CIS-Fehlertexte mit den dazugehörigen Erläuterungen werden auf SYSLST ausgedruckt.

```
LSTI
```

Informationstexte ausdrucken.

Alle Informationstexte mit den dazugehörigen Erläuterungen werden auf SYSLST ausgedruckt.

```
LSTM[_{Name/Teilname*}]
```

Masken ausdrucken.

`Name`	Name der Maske, ab der aufgelistet werden soll.
`Teilname*`	Teil eines Namens (mindestens 1 Byte). Ab diesem Teilnamen, ergänzt mit Spaces, werden die Masken aufgelistet.

Ohne Angabe des Namens bzw. Teilnamens werden alle Masken aufgelistet.

Pro Maske wird eine Seite auf SYSLST ausgegeben.

Am Ende wird die Anzahl der aufgelisteten Elemente ausgegeben.

```
NOP
```

Null-Operation.

Die NOP-Anweisung hat keine Wirkung.

Wird die NOP-Anweisung als erstes Kommando benützt, laufen Initialisierungsroutinen (z.B. Einrichten der DABEL, falls noch nicht vorhanden) ab.

PROC

Unterdrückung von Ausgaben bei Programmen, die in einer Prozedur ablaufen.

Ausgaben wie - Parameteranforderung oder die
- Meldung, daß der EDT aufgerufen wird,

werden unterdrückt.

Diese Funktion wird mit CM=IM99 quittiert.

PUTD

Datenbeschreibung generieren (unbedingt).

Alle im EDT-Speicher (Arbeitsdatei-0) stehenden Datenbeschreibungen werden in die geöffnete DABEL geschrieben.

Ist eine Datenbeschreibung schon in der DABEL vorhanden, so wird sie überschrieben.

Sind im EDT-Speicher auch Masken vorhanden, so werden diese ignoriert.

Jede Datenbeschreibung wird auf SYSLST protokolliert.

PUTM

Bildschirmmasken generieren (unbedingt).

Alle im EDT-Speicher (Arbeitsdatei-0) stehenden Masken werden in die geöffnete DABEL geschrieben.

Ist eine Maske schon in der DABEL vorhanden, so wird sie überschrieben.

Sind im EDT-Speicher auch Datenbeschreibungen vorhanden, so werden diese ignoriert.

Jede Maske wird auf SYSLST protokolliert.

```
READ_Dateiname
```

SAM-Datei lesen

`Dateiname` Name der Datei, die gelesen wird.

Das Kommando wird in der Form: @READ'Dateiname' an den EDT übergeben.

Die gelesenen Sätze werden in den EDT-Speicher (Arbeitsdatei-0) geschrieben. Der Speicher wird fortgeschrieben.

Soll ab Anfang des Speichers geschrieben werden, muß vorher mit dem RESET-Kommando (vgl. Seite 66) der EDT-Speicher gelöscht werden.

Nach Ausführung des READ-Kommandos wird die Anzahl der Datenbeschreibungen und Masken protokolliert.

Wird eine ISAM-Datei mit dem READ-Kommando gelesen, gibt der EDT die Meldung ISAM FILE - DOING @GET aus.

```
RECF_{F/V}
```

Vorgabe des Satztyps (RECFORM).

Mit dem RECF-Kommando kann vor Ausführung der LDB-Anweisung (vgl. Seite 51) der Satztyp für die LEASY-Datei vorgegeben werden. Der Befehl ist anzuwenden wenn die LDB-Anweisung mit SHOW-FILE-ATTRIBUTES den Satztyp nicht feststellen kann, d.h. es ist nur der LEASY-Katalog und nicht die Datei vorhanden, bzw. wenn es sich bei der LEASY-Datei um eine PAM-Datei handelt.

RESET

EDT-Speicher löschen.

Löscht die Arbeitsdateien 0 und 1 des EDT-Speichers.

Ebenso löschen die Kommandos ADB und LDB den EDT-Speicher.

```
RGEND[_{Name/Teilname*/*}]
```

Regenerieren der Datenbeschreibungen der geöffneten DABEL.

`Name`	Name einer zu regenerierenden Datenbeschreibung.
`Teilname*`	Alle Datenbeschreibungen, deren Namen mit "Teilname" beginnen, werden regeneriert .
`*`	Alle Datenbeschreibungen sollen regeneriert werden.

Aus der geöffneten DABEL werden die Datenbeschreibungen regeneriert, d.h. in den EDT-Speicher (Arbeitsdatei-0) geschrieben.

Der EDT-Speicher wird fortgeschrieben, er wird nicht gelöscht. Datenbeschreibungen und Masken, die bereits im Speicher stehen, werden nicht gelöscht. (Löschen des EDT-Speichers: Vgl. RESET-Kommando Seite 66).

Nach Ausführung des RGEND-Kommandos wird die Anzahl der im EDT-Speicher stehenden Datenbeschreibungen protokolliert.

```
RGENM[_{Name/Teilname*/*}]
```

Regenerieren der Masken der geöffneten DABEL.

`Name`	Name einer zu regenerierenden Maske.
`Teilname*`	Alle Masken, deren Namen mit "Teilname" beginnen, werden regeneriert .
*	Alle Masken werden regeneriert .

Aus der geöffneten DABEL werden die Masken regeneriert, d.h. in den EDT-Speicher (Arbeitsdatei-0) geschrieben.

Der EDT-Speicher wird fortgeschrieben, er wird nicht gelöscht. Datenbeschreibungen und Masken, die bereits im Speicher stehen, werden nicht gelöscht. (Löschen des EDT-Speichers: Vgl. RESET-Kommando Seite 66).

Nach Ausführung des RGEND-Kommandos wird die Anzahl der im EDT-Speicher stehenden Masken protokolliert.

```
SAVE_Dateiname
```

Schreiben einer ISAM-Datei.

`Dateiname` Name der Datei, in die alle Sätze aus dem EDT-Speicher (Arbeitsdatei-0) geschrieben werden.

Interner Ablauf: `@UNSAVE'Dateiname'`

`@SAVE'Dateiname'`

Ist die Datei noch nicht vorhanden, so wird bei @UNSAVE die Meldung:

```
FILE NOT IN CATALOG
```

ausgegeben.

```
SDB_Name[,PRE=nn][,GRP=xxxx][,BEG=nn]
```

Generierung von D- und E-Segmenten aus einer LMS-Datei.

Name	Name des Elements, aus dem D- und E-Segmente generiert werden.	
PRE=nn	Präfix-Länge der Namen in der LMS-Datei (1- oder 2-stellig). Dieses Präfix wird nicht als Name in das E-Segment übernommen. Standardwert: PRE=00	
GRP=xxxx	Angabe eines Abschnittnamens (MV-Format). Fehlt der Operand GRP werden Sätze mit V-Format beschrieben.	
BEG=nn	Angabe einer Anfangsadresse (1- oder 2-stellig):	BEG=05 für V-Format BEG=09 für MV-Format

Diese Anweisung bezieht sich auf die aktuelle LMS-Datei, die mit der SRC-Anweisung (vgl. Seite 74) eröffnet wurde.

Es werden D- und E-Segmente generiert und in den EDT-Speicher (Arbeitsdatei-0) geschrieben. Der EDT-Speicher wird fortgeschrieben (zum Rücksetzen/ Löschen, vgl. RESET-Kommando Seite 66). Das MV-Format kann generiert werden. Für einen Abschnitt wird ein COBOL-COPY-Element benötigt.

Die D- und E-Segmente stehen im EDT-Speicher 0, die COBOL-Statements im EDT-Speicher 1.

Vgl. auch Seite 48: Bearbeiten eines COBOL-COPY-Elements.

Hinweis: Der Typ der LMS-Datei muß OSM sein, damit die LMS-Datei von CISGEN bearbeitet werden kann. Ist dies nicht der Fall, so kann das COBOL-COPY-Element mit der SEL-Anweisung (vgl. LMS-Benutzerhandbuch) aus der LMS-Datei in eine SAM-/ISAM-Datei gebracht werden und mit der IDB-Anweisung (vgl. Seite 47) bearbeitet werden.

```
SEG_aaaa,lll[,fff],Name
```

Generieren von D- und E-Segmenten.

`aaaa` Relative Adresse des Feldes (1- bis 4-stellig)

`lll` Länge des Feldes (max. 255) (1- bis 3-stellig)

`fff` Folgende 3 Bytes können angegeben werden:

- logische Feldbedeutung: O, W, V, P, U (vgl. Seite 95)
- Dezimalstellen : 0, 1, ... , F
- Darstellung : P, B, R, L, A, T (vgl. Seite 96)

Werden weniger als 3 Bytes angegeben, so wird rechtsbündig gespeichert.

Standardwert: T

`Name` Name des Feldes (1 bis 15 Bytes)

Das D- und E-Segment werden in den EDT-Speicher (Arbeitsdatei-0) geschrieben.

```
SINF[_{Name/Teilname*}]
```

Inhaltsverzeichnis der LMS-Datei ausgeben.

Name	Name eines Elements in der LMS-Datei, ab dem das Inhaltsverzeichnis auszugeben ist.
Teilname*	Teil eines Namens (mindestens 1 Byte). Ab diesem Teilnamen, ergänzt mit Spaces, wird das Inhaltsverzeichnis ausgegeben.

Ausgabe:

1. Zeile:	FILE Dateiname
2. Zeile:	NAME VER/DATE
3.-20. Zeile:	18 Elemente des LMS-Datei
21. Zeile:	WEITER (J/N)

NAME	Name des Elements
VERS/DATE	Version/Datum
WEITER (J/N)	J: Auflistung der nächsten 18 Elemente. N: Beenden der Funktion.

SORT

Sortieren einer Datenbeschreibung nach Adressen.

Die im EDT-Speicher stehende Datenbeschreibung wird aufsteigend sortiert nach:

- Abschnittsnamen
- Adressen

Es darf nur eine Datenbeschreibung im EDT-Speicher stehen.

```
SRC_Dateiname
```

Eröffnen einer LMS-Datei.

`Dateiname` Name der LMS-Datei (DVS-Name)

Ist schon eine LMS-Datei eröffnet, so wird diese zuerst geschlossen. Es kann immer nur eine LMS-Datei offen sein.

Die Datei wird zum Lesen eröffnet.

Es muß eine ISAM-Datei mit KEYPOS=5 und KEYLEN=8 sein.

SYSDTA

EDT-Eingabe auf SYSDTA setzen.

Setzt das Eingabemedium für EDT-L fest.

SYSDTA bewirkt, daß EDT-L über RDATA liest, also von der logischen Datei SYSDTA.

```
TERMINAL
```

EDT-Eingabe auf TERMINAL setzen.

Setzt das Eingabemedium für EDT-L fest (vgl. SYSDTA Seite 75).

TERMINAL bewirkt, daß EDT-L über WRTRD liest, also über die Datenstation.

UREAD

Satz aus dem EDT-Speicher lesen.

Kann nur benützt werden, wenn CISGEN als Unterprogramm aufgerufen wird.

Die Adresse des Bereichs, wohin der Satz gelesen wird, steht im 5. Parameter. Der Satz hat V-Format und ist maximal 256 Bytes lang.

Nach dem Lesen des Satzes wird der EDT-Lesezeiger auf den nächsten Satz positioniert.

Folgende Meldungen sind möglich:

`CM = IM00`	Satz wurde zur Verfügung gestellt.
`CM = IM01`	Dateiende, es wurde kein Satz gelesen.

Der EDT-Lesezeiger kann außer auf den ersten Satz des EDT-Speichers (vgl. Seite 78) nicht gezielt positioniert werden.

Beim ersten UREAD-Aufruf bzw. beim ersten UREAD-Aufruf nach einer RESET-Anweisung wird der erste Satz gelesen.

USETB

Lesezeiger des EDT auf Anfang setzen.

Beim nächsten UREAD (vgl. Seite 77) wird der erste Satz aus dem EDT-Speicher gelesen.

```
UWRITE
```

Satz an CISGEN übergeben.

Kann nur benützt werden, wenn CISGEN als Unterprogramm aufgerufen wird.

Die Adresse des Satzes steht im 5. Parameter. Der Satz muß ein Satzlängenfeld enthalten und darf maximal 256 Bytes lang sein.

Der Satz wird in den EDT-Speicher übernommen.

```
WHY cm
```

Codierte Meldungen erläutern.

`cm` Codierte Meldung (4 Bytes)

Es werden die Erläuterungen zur angegebenen codierten Meldung ausgegeben.

```
WRITE_Dateiname
```

Schreiben einer SAM-Datei.

`Dateiname` Name der Datei, in die alle Sätze aus dem EDT-Speicher geschrieben werden.

Interner Ablauf: `@UNSAVE'Dateiname'`

`@WRITE'Dateiname'`

Ist die Datei noch nicht vorhanden, so wird bei @UNSAVE die Meldung:

`FILE NOT IN CATALOG`

ausgegeben.

2.4 Satzbeschreibung - Datenbankdefinition

Die Datenbankdefinition beinhaltet:

- Satzbeschreibung
- Definition des Datenbankschlüssels
- Definition der Suchwege
- Definition der Dateinamen
- Definition der Zugriffsberechtigung

Die Anweisungen dazu werden in der Satzbeschreibung, wie auf den folgenden Seiten beschrieben, angegeben und mit CISGEN in der Datenbeschreibungsdatei (DABEL) abgespeichert.

2.4.1 A-Segment der Satzbeschreibung

Das A-Segment enthält den Namen der Datenbeschreibung, der zugleich als Paßwort verwendet wird, und Steuerfunktionen. Das A-Segment wird bei Erstellung der Satzbeschreibung als erste Anweisung angegeben.

Stellen	Inhalt	Bedeutung
1 - 2	Space	Reserviert
3 - 3	A	Segmentname
4 - 9	Name	Datenbeschreibungsname. Der 6 Bytes lange Name (Buchstaben und/oder Ziffern, aber keine Spaces sind erlaubt) muß angegeben werden und wird als Paßwort für den Datenzugriff verwendet.
	oder Spaces	falls die auf den Namen folgenden Parameter nicht in einer Zeile eingegeben werden (können), bleibt das Namensfeld ab der zweiten Zeile leer.
10 - 80	Parameter	Schlüsselwortparameter im Format: Parameter1=Wert1[,Parameter2=Wert2,...] Falls die Parameter in mehreren Zeilen eingegeben werden, muß ein Parameter innerhalb einer Zeile vollständig erfaßt und durch Komma abgeschlossen sein. In der Folgezeile bleibt das Namensfeld leer.

Parameter im A-Segment

Parameter - Typ	Parameter	Parameterwert	Standardwert
Dateiparameter	HD	Dateiname	
	VD	Dateiname	
	HDLINK	Linkname für HD	
	VDLINK	Linkname für VD	
	EOC	N/J	N
Steuerungsparameter	EK	Zeichenfolge	
	TK	Zeichenfolge	
	LOGADR	N/J	N
	MAXDES	0 ≤ nnn ≤ 248	
	SQLTAB	N/J	N
	TYP	C/L	C
Datenschutzparameter	DSA	N/J	N
	DSS	N/J	N
	DSU	N/J	N
Ausgabeparameter	FORM	nn	72
	KOPF	Text (1-30 Zeichen)	
	ZEILEN	nn/m	50/0
	ZEIT	N/J	N
Katalogparameter	BLKSIZE	(STD,b)	(STD,1)
	RECFORM	V/F	V
	RECSIZE	r	
	SPACE	(p,s)	(3*b,3*b)
	HDPOOLLNK	Linkname	
	VDPOOLLNK	Linkname	
	HDBLKCTRL	DATA	
	VDBLKCTRL	DATA	

Dateiparameter

Die Dateiparameter steuern das Öffnen und Schließen der Haupt- und der Verweisdatei. In der Regel werden OPEN und CLOSE verdeckt ausgeführt (impliziter OPEN / CLOSE). Es werden die Dateien, die mit HD=...bzw. VD=... angegeben sind, angesprochen. Der Anwender muß in diesem Fall die Dateinamen nicht kennen und kann deshalb nicht direkt, d.h. ohne CIS, auf die Daten zugreifen (Datenschutz).

HD=Dateiname — Name der Hauptdatei. Fehlt dieser Parameter, muß EOC=J angegeben werden.

VD=Dateiname — Name der Verweisdatei. Der Parameter kann entfallen, wenn keine Invertierungen verwendet werden sollen (vgl. D-Segment SF1: Seite 97). Fehlt dieser Parameter, obwohl Invertierungen auftreten, muß EOC=J angegeben werden.

HDLINK=Linkname — Zuweisung der Hauptdatei über einen Linknamen. Vor Aufruf des Programms muß die Datei mit dem /SET-FILE-LINK Kommando zugewiesen werden. Dieser Parameter kann alternativ zu HD=Dateiname angegeben werden.

VDLINK=Linkname — Die Ausführungen zu HDLINK=Linkname gelten entsprechend.

EOC=N/J
- N: Ein expliziter OPEN oder CLOSE darf nicht ausgeführt werden (Standardwert).
- J: Ein expliziter OPEN oder CLOSE darf ausgeführt werden, auch dann, wenn HD und/oder VD angegeben sind.

Hinweise:

a) Wird bei der HD bzw. bei der VD mit einer Datei-Generationsgruppe gearbeitet, so ist dies nur mit einer absoluten Generationsnummer *nnnn ($1 \leq nnnn \leq 9999$) möglich, wobei führende Nullen angegeben werden müssen. Relative Generationsnummern sind unzulässig.

   (Beispiel: HD=HD.PERSB1(*0047))

b) Userid/Catid können bei den Dateinamen angegeben werden.

Steuerungsparameter

Die Steuerungsparameter steuern einzelne Funktionen beim Update und Laden der Datenbank.

LOGADR=N/J N: Beim Einfügen oder Laden von Sätzen in die ISAM-HD wird von CIS kein Datenbankschlüssel (Ordnungsbegriff) vergeben. Der Anwender ist dafür verantwortlich, daß der übergebene Satz einen eindeutigen Ordnungsbegriff enthält. (Standardwert)

J: Beim Einfügen oder Laden von Sätzen in die ISAM-HD wird von CIS ein Datenbankschlüssel (Ordnungsbegriff) vergeben. In Abhängigkeit der Definition des Ordnungsbegriffs (vgl. D-Segment - Seite 95) wird ein 1 bis 4 Bytes langer binärer Ordnungsbegriff vergeben. Der erste Wert ist 1, die Schrittweite 1.

MAXDES=nnn MAXDES ist die Länge des längsten zu invertierenden Feldinhalts (vgl. D-Segment SF1 - Seite 97). nnn ist eine dreistellige Zahl; führende Nullen entfallen. Der Wert kann zwischen 0 und 238 abzüglich der Länge des Ordnungsbegriffes liegen.

EK=Z1Z2...ZnE Stichwortinvertierung - Vollinvertierung: Dieser Parameter bewirkt die Invertierung von Stichworten eines in einem Datenfeld stehenden Textes (vgl. D-Segment SF1 - Seite 97). Sämtliche Stichworte des betreffenden Textes werden invertiert.

In der Parametervariablen werden Zeichen (Z1Z2...Zn) festgelegt, die als Trenner für die Stichwortextraktion gelten (Feldanfang und -ende werden stets als Trenner behandelt). Die Zeichenfolge hat keinen Zwischenraum und wird durch E begrenzt, d.h. es sind alle Zeichen mit Ausnahme von E zugelassen. Ein eventueller Zwischenraum wird als entsprechender Trenner (Space) gewertet.

Es darf kein TK-Parameter angegeben sein.

Beispiele: EK=,-E sei gegeben.

Mit dem Feldinhalt "MEIER,HANS-PETER" werden folgende Stichworte invertiert:

1. MEIER
2. HANS
3. PETER

Dagegen werden mit dem Feldinhalt "MEIER HANS-PETER" nur 2 Stichworte gebildet (Space ist nicht als Trenner definiert worden) und invertiert:

1. MEIER HANS
2. PETER

TK=Z1Z2...ZnE Stichwortinvertierung - Teilinvertierung: Dieser Parameter bewirkt die Invertierung von Stichworten eines in einem Datenfeld stehenden Textes (vgl. D-Segment SF1 - Seite 97). Die Zeichenfolge (Z1Z2...Zn) hat keinen Zwischenraum und wird durch E begrenzt, d.h. es sind alle Zeichen mit Ausnahme von E zugelassen. Ein eventueller Zwischenraum wird als entsprechender Trenner (Space) gewertet. Jedes angegebene Zeichen (Z1Z2...Zn) schaltet die Stichwortinvertierung beim 1. Auftreten ein und beim 2. Aufreten wieder aus, beim 3. Auftreten ein und beim 4. Aufreten wieder aus, usw.. (Feldanfang und -ende werden nicht als Trenner behandelt)

Es darf kein EK-Parameter angegeben sein.

Beispiele: TK=,-E sei gegeben.

Mit dem Feldinhalt "MEIER,HANS-PETER,8000-MUENCHEN-40" werden folgende Stichworte invertiert:

1. Hans
2. 8000

Dagegen werden mit dem Feldinhalt "MEIER,HANS-PETER,8000-MUENCHEN-40," folgende Stichworte invertiert:

1. Hans
2. 8000
3. 40

Mit dem Feldinhalt ",MEIER,HANS-PETER,8000-MUENCHEN-40" werden dagegen folgende Stichworte invertiert:

1. MEIER
2. PETER
3. MUENCHEN

SQLTAB=J/N Dieser Parameter dient dazu, Multiplitäten in CIS-Datenbanken zu linearisieren. Die betreffende Datenbeschreibung muß den Ordnungsbegriff (bei MV-Format mit richtiger Abschnittsart) und darf bei MV-Format nur e i n e Abschnittsart, bei Wiederholfeldern (V- oder MV-Format) nur das betreffende Feld beschreiben.

Beim Ansprechen einer solchen Satzbeschreibung mit dem DBH werden für das Wiederholfeld ein Feld Feldmultiplität und für den Wiederholabschnitt ein Feld Abschnittsmultiplität zusätzlich erzeugt.

Beim Suchen mit einer derartigen Beschreibung wird von der ZPL nicht nur der Ordnungsbegriff (Isam-Key) des betreffenden Satzes sondern auch die Multiplität (AM/FM) des Treffers abgelegt.

Beispiel: (Vgl. Kommando SUCHEN im Manual 3)

SEGM	KOPFBESCHREIBUNG												
	NAME	PARAMETER IM A-SEGMENT											
AA	KUNDEN	LOGADR=J,SQLTAB=J,DSS=J,DSA=J,DSU=J,HD=HD.KUNDEN											
FELDBESCHREIBUNG													
SEGM	QUELL-DEFINITION			LOG	DARST	TV	ZIEL-DEFINITION			SONDER FUNKTION		WA	SEGM	FELDBEZEICHNUNG TRANSFORMATIONS-ERGAENZUNG
	AA	ADR	LNG				AA	ADR	LNG	SF1	SF2			
3	4	8	12	15		18	21	25	29	32	36	0	5	46 (41-44 FREI)
DDD	PERS	9	3	O	B								EEE	EDV-NR
DDD	WARE	9	5		T							WWW	EEE	ARTIKELNR
DDD	WARE	14	9		2R								EEE	PREIS
DDD	WARE	23	34		T								EEE	BEZEICHNUNG

E N D E

Bei Datenbeschreibungen mit Wiederholabschnitten und Wiederholfeldern werden im geladenen Zustand für die Abschnitts- und Feldmultiplität zwei Zählfelder (binär, Länge=2) erzeugt. Die Bezeichnung dieser Felder ist AM bzw.FM. D.h. die Feldbezeichnungen AM und FM sind für die Zählfelder reserviert und dürfen somit in der Datenbeschreibung nicht vergeben werden! Die logische Feldbedeutung ist A bzw. F.

Die Zählfelder können mit dem Kommando ZEIGE,A ausgegeben werden:

Beispiel für Wiederholabschnitt:

```
FELD             AART W  ADR SEGM LNG T DARSTELLUNG DZ

EDV-NR           PERS      9         3 O BINAER
ARTIKELNR        WARE W    9         5   ZEICHEN LB
PREIS            WARE W   14         9   ENTPACKT RB
BEZEICHNUNG      WARE W   23        34   ZEICHEN LB
AM               WARE W    1         2 A BINAER
```

Beispiel für Wiederholfeld:

```
FELD             AART W  ADR SEGM LNG T DARSTELLUNG DZ

EDV-NR                     9         3 O BINAER
ARTIKELNR                  9         5   ZEICHEN LB
PREIS                     14         9   ENTPACKT RB
BEZEICHNUNG               23        34 W ZEICHEN LB
FM                        23         2 F BINAER
```

Beispiel für Wiederholfeld im Wiederholabschnitt:

```
FELD             AART W  ADR SEGM LNG T DARSTELLUNG DZ

EDV-NR           PERS      9         3 O BINAER
ARTIKELNR        WARE W    9         5   ZEICHEN LB
PREIS            WARE W   14         9   ENTPACKT RB
BEZEICHNUNG      WARE W   23        34 W ZEICHEN LB
AM               WARE W    1         2 A BINAER
FM               WARE W   23         2 F BINAER
```

TYP=C/L C: Die Verweisdatei ist/wird von CIS erzeugt (Standardwert).
L: Es handelt sich um eine LEASY-SI-Datei. Die Sekundärindizes sind in einer LEASY-SI-Datei gespeichert. Sie kann von CIS nur gelesen werden, ein Update ist nicht möglich.

Datenschutzparameter

Die Datenschutzparameter steuern den Zugriff auf bestimmte Daten. Such-, Ausgabe- und Updatefunktionen können mit Datenbeschreibungen nur ausgeführt werden, wenn entsprechende Parameter gesetzt sind.

Jede Datenbank kann beliebig oft beschrieben werden. Die einzelnen Benutzer können mit dem jeweiligen Paßwort nur die Zugriffsarten ausführen, die für sie zugelassen sind. Ebenso können die einzelnen Anwender nur auf die Felder des Datenbestandes zugreifen, die in den für sie eingerichteten Satzbeschreibungen (geschützt durch das Paßwort) angegeben sind. Damit ist ein feld- und funktionsbezogener Datenschutz realisierbar (vgl. Bild 1 und vgl. Beispiel 2 - Seite 102).

DSS=N/J
N: Die Datenbeschreibung darf nicht für Suchen verwendet werden. (Standardwert)
J: Die Datenbeschreibung darf für Suchen verwendet werden.

DSA=N/J
N: Die Datenbeschreibung darf nicht für Ausgaben verwendet werden. (Standardwert)
J: Die Datenbeschreibung darf für Ausgaben verwendet werden.

DSU=N/J
N: Die Datenbeschreibung darf nicht für Update verwendet werden. (Standardwert)
J: Die Datenbeschreibung darf für Update verwendet werden.

Bild 1: Schematische Darstellung der durch Datenbank-Paßworte beschränkten Zugriffsmöglichkeiten auf Felder in der CIS-Datenbank.

Datei	Satzbeschreibung 1	Satzbeschreibung 2	Satzbeschreibung 3
	UPDATE DSU=J	SUCHEN DSS=J	AUSGABE DSA=J
Feld 1	Feld 1		Feld 1
Feld 2		Feld 2	
Feld 3	Feld 3	Feld 3	
Feld 4			
Feld 5	Feld 5	Feld 5	
Feld 6		Feld 6	Feld 6

Mit Satzbeschreibung 1 können nur die Felder 1, 3, und 5 verändert werden.
Mit Satzbeschreibung 2 kann nur nach den Feldern 2, 3, 5, und 6 gesucht werden.
Mit Satzbeschreibung 3 können nur die Felder 1 und 6 ausgegeben werden.

Ausgabeparameter

Die Ausgabeparameter steuern die Form der Listen, die mit dem Drucke-Kommando von CIS tabellarisch oder zeilenweise erstellt werden können (vgl. Beispiel 3 - Seite 103).

KOPF=Text		Text ist eine 1 bis 30 Zeichen lange Konstante, die als Seitenüberschrift auf die Listen gedruckt wird.
ZEILEN=nn/m	nn	Anzahl der Zeilen je Druckseite. nn ist zweistellig. Wenn der Wert nicht angegeben ist, wird 50 angenommen.
	m	Anzahl der Leerzeilen zwischen den einzelnen Datenzeilen. m ist einstellig. Wenn der Wert nicht angegeben ist, wird 0 angenommen, d.h. es werden keine Leerzeilen gedruckt.
FORM=nn	nn	Anzahl der Zeilen der Titelseite (TEXT-Kommando). Wird der Parameter nicht angegeben, wird 72 angenommen.
ZEIT=N/J	N:	In der Seitenüberschrift wird die Uhrzeit nicht angegeben. (Standardwert).
	J:	In der Seitenüberschrift wird bei tabellarischer Druckausgabe die Uhrzeit im Format hh.mm angegeben.

FILE-Parameter

CIS generiert das CREATE-FILE und SET-FILE-LINK Kommando für HD und VD wenn über die Kommandos EI,K bzw. PUT,K eine neue Datei erstellt wird. Dabei können einige Parameter für das CREATE-FILE und SET-FILE-LINK Kommando im A-Segment angegeben werden (vgl. Beispiel 3 Seite 103).

BLKSIZE=(STD,b)	Angabe des Blockungsfaktors b ($1 \leq b \leq 16$) für die Hauptdatei. Fehlt dieser Parameter, so wird Blockungsfaktor 1 angenommen.
RECFORM=f	Satztyp: f = F Die HD besteht aus Sätzen mit fester Länge. f = V Die HD besteht aus Sätzen mit variabler Länge. Fehlt dieser Parameter so wird RECFORM=V angenommen.
RECSIZE=r	r = Satzlänge in der HD. Dieser Parameter muß bei RECFORM=F angegeben werden.
SPACE=(p[,s])	p = Primärzuweisung der HD. Fehlt dieser Parameter, so wird der dreifache Blockungsfaktor angenommen. s = Sekundärzuweisung der HD. Fehlt dieser Parameter, so wird der dreifache Blockungsfaktor angenommen.
HDPOOLLNK= VDPOOLLNK= } Linkname	Linknamen, mit denen die HD oder VD an einen definierten ISAM-POOL angeschlossen werden kann.
HDBLKCTRL= VDBLKCTRL= } DATA	Der Parameter HD/VDBLKCTRL=DATA muß bei der Einrichtung von NK-Dateien auf Key-Platten angegeben werden.

Sollen bei der Anlage einer Haupt-, bzw Verweisdatei zusätzliche Parameter zu den bei der automatischen Anlage von CIS (vgl. Seite 92 und 93) vergebenen notwendig sein (z.B. Angabe einer bestimmten Platte), so muß die HD bzw. VD manuell mit dem CREATE-FILE Kommando angelegt werden.

Beispiel:

```
/CREATE-FILE FILE-NAME=Name[SUPPORT=PRIVATE-DISK(VOLUME=V,
DEVICE-TYPE=d,SPACE=RELATIVE(PRIMARY-ALLOCATION=p,
SECONDARY-ALLOCATION=s))]
```

Eine Angabe von BLKSIZE, RECFORM, und RECSIZE mit dem SET-FILE-LINK Kommando erübrigt sich bei der manuellen Anlage, da diese Parameter auf jeden Fall von CIS vergeben werden.

Automatische Anlage der Hauptdatei durch CIS:

Ist keine HD vorhanden, so wird vor dem ersten Schreibversuch von CIS ein CREATE-FILE und SET-FILE-LINK Kommando mit nachstehenden Parametern abgesetzt:

Parameter in den BS2000-Kommandos	Möglicher Parameter im A-Segment der Datenbeschreibung	Wert wenn der Parameter im A-Segment fehlt
SPACE=RELATIVE	SPACE=(p[,s]) 1)	primär: 3 x BLKSIZE sekundär: 3 x BLKSIZE
LINK-NAME	Wird von CIS automatisch vergeben.	CIS-intern vergeben
FILE-NAME	HD=Name	
ACCESS-METHOD	Wird von CIS automatisch vergeben.	ISAM
KEY-LENGTH	Wird von CIS automatisch vergeben.	Länge OB im D-Segment
KEY-POSITION	Wird von CIS automatisch vergeben.	V- Format: Adresse des OB im D-Segment MV-Format: Adresse des OB + 4 im D-Segment
BUFFER-LENGTH	BLKSIZE=(STD,1 ≤ b ≤ 16)	V-Format: (STD,1) MV-Format: (STD,1)
RECORD-FORMAT	RECFORM=F (nur wenn RECSIZE=r)	V
RECORD-SIZE	RECSICE=r (nur wenn RECFORM=F)	0

1) Die Angabe des Parameters SPACE im A-Segment ist nur dann sinnvoll, wenn sicher ist, daß die Größe der HD die dreifache BLKSIZE übersteigt. Die primäre (p) und sekundäre (s) Speicherplatzzuweisung sollte sinnvollerweise einem Vielfachen von (3 x BLKSIZE) entsprechen.

Automatische Anlage der Verweisdatei durch CIS:

Ist keine VD vorhanden, so wird vor dem ersten Schreibversuch von CIS ein CREATE-FILE und ein SET-FILE-LINK Kommando mit nachstehenden Parametern abgesetzt:

Parameter in den BS2000-Kommandos	Wert
SPACE=RELATIVE	(PRIMARY-ALLOCATION=12,SECONDARY-ALLOCATION=12)
LINK-NAME	CIS-intern vergeben
FILE-NAME	im A-Segment
ACCESS-METHOD	ISAM
PADDING-FACTOR	0
KEY-LENGTH	MAXDES + 7
KEY-POSITION	5
BUFFER-LENGTH	(STD,1)
RECORD-FORMAT	V
RECORD-SIZE	0

Mehrbenützbarkeit der Dateien:

Wird mit einem unabhängigen CISKOOR gearbeitet, so wird für die HD und die VD noch folgendes Kommando von CIS erzeugt:

```
/MODIFY-FILE-ATTRIBUTES FILE-NAME=Name,PROTECTION=PARAMETERS
(ACCESS=WRITE),USER-ACCESS=ALL-USERS
```

2.4.2 D-Segment der Satzbeschreibung

Das D-Segment beschreibt die einzelnen Felder des Satzes und definiert für sie den Zugriffspfad für Suchoperationen.

Stellen	Bezeichnung	Bedeutung
1 - 2	reserviert	Enthält Spaces.
3 - 3	Segment	Enthält D.
4 - 7	Abschnittsname	Bei Sätzen mit V-Format bleibt dieses Feld leer. Bei Sätzen mit MV-Format wird der Abschnittsname vierstellig eingetragen. Bei der Definition des Überbegriffes eines Compound-Keys (zusammengesetzter Key) bleibt dieses Feld leer. Nur bei den Unterfeldern des Compound-Keys wird bei Sätzen mit MV-Format der Abschnittsname der Felder eingetragen. (vgl. Beispiel 5 - Seite 105 und Beispiel 6 - Seite 106)
8 - 11	Adresse	Rechtsbündiger Eintrag der relativen Adresse eines Feldes innerhalb eines Satzes (bei V-Format) bzw. eines Abschnitts (bei MV-Format). Führende Nullen können entfallen. Die relative Adresse des erstmöglichen Feldes ist formatabhängig: V-Format: 5 MV-Format: 9 (gilt auch bei RECFORM=F) Dieses Feld bleibt bei der Definition des Überbegriffes eines Compound-Keys leer. Bei der Definition der Unterfelder eines Compound-Keys, wird die relative Adresse des verwendeten Feldes angegeben.
12 - 14	Länge in Bytes	Die Länge des beschriebenen Feldes, bzw. die Summe der Längen aller Unterfelder eines Compound-Keys wird rechtsbündig eingetragen. Bei einem variabel langen Feld wird die maximale Länge eingetragen (vgl. Feldbedeutung V - Seite 95). Führende Nullen können entfallen. CIS ignoriert ein Feld wenn es kürzer als die angegebene Länge (z.B. letztes Feld) und nicht variabel lang ist (Feldbedeutung ungleich V). Bei der Ausgabe (z.B. ZEIGEN) stehen Blanks. Die maximale Länge ist abhängig von der jeweiligen Feldbedeutung:

Darstellung	max. Länge in Bytes
P	8
H	7
B	4
R	15
L	255
A	255
T	255

Stellen	Bezeichnung	Bedeutung
15 - 17	Feldbedeutung	
15 - 15	Log. Feldbedeutung	K - Komprimiertes Feld.

Feld variabler Länge, für das ein Längenzähler geführt wird (0 - 255). Die Ausführungen zum Wiederholfeld gelten auch hier, es können aber mehrere K-Felder pro Satz oder Abschnitt definiert werden. Alle haben die letzte feste Adresse als Anfangsadresse und unterscheiden sich durch eine fortlaufende Nummer in Spalte 25 - 28.

O - Ordnungsbegriff.

Das beschriebene Feld ist der Ordnungsbegriff (Datenbank-schlüssel). Für den gesamten Satz darf nur ein Ordnungsbegriff vergeben werden. Falls der Satz MV-Format hat, muß der Ordnungsbegriff im ersten Abschnitt definiert sein. Die logische Feldbedeutung O definiert den (einzigen) direkten Suchpfad. Wird der ISAM-Schlüssel von CIS vergeben, also LOGADR=J, so muß dieses Feld binär mit der Länge 1 bis 4 definiert werden.

P - Pseudofeld.

Das Pseudofeld ist nicht Bestandteil des Satzes. Pseudofelder können zum Sichern von Zielpunktlisten in der Verweisdatei verwendet werden (vgl. SICHERN-Kommando - Manual 3). Für die Definition des Pseudofeldes sind folgende Felder der Satzbeschreibung auszufüllen:

- Länge	Stelle 12 - 14
- Feldbedeutung	Stelle 15 - 17
- Kurzname	Stelle 32 - 34
- E-Segment	Stelle 45 - 60

S - Pseudofeld für sortierte Zielpunktlisten.

Es gelten die Ausführungen wie bei der logischen Feldbedeutung P. Im Unterschied zu P-Feldern kann mit einem S-Feld eine sortierte Zielpunktliste (ZPL) in der Verweisdatei gesichert werden.

U - Unterfeld bzw. unterteiltes Feld.

Falls ein Feld unterteilt (redefiniert) werden soll, müssen sowohl der Überbegriff als auch die Unterfelder mit der logischen Feldbedeutung U definiert werden.

V - Feld variabler Länge.

Ein Feld mit variabler Länge muß beim V-Format als letzes Feld des Satzes, beim MV-Format als letztes Feld des Abschnitts definiert werden. Es darf für einen Satz (bei V-Format) bzw. für einen Abschnitt nur ein Feld mit variabler Länge definiert werden. Im Längenfeld wird die maximale Länge des Feldes eingetragen.

Stellen	Bezeichnung	Bedeutung
		W - Wiederholfeld. Ein Wiederholfeld muß beim V-Format als letzes Feld des Satzes, beim MV-Format als letztes Feld des Abschnitts definiert werden. Es darf für einen Satz (bei V-Format) bzw. für einen Abschnitt nur ein Wiederholfeld definiert werden. Z - Zusammengesetztes Feld (COMPOUND-KEY). Der Überbegriff und die Unterfelder eines Compound-Keys werden mit 'Z' gekennzeichnet. Beim MV-Format müssen alle Unterfelder, aus denen der Compound-Key besteht, aus demselben Abschnitt stammen. Die Quellfelder, d.h. die Felder, die durch die Unterfelder 'redefiniert' werden, müssen in Spalte 15 'U' enthalten.
16 - 17	Darstellung	In diesem Feld wird eine Zahl und ein Zeichen bzw. Space und ein Zeichen eingetragen. Folgende Darstellungen sind zugelassen: nP Gepackte Zahl mit n Dezimalstellen. nH Gepackte Zahl ohne Vorzeichen mit n Dezimalstellen. B Binäre Zahl. nR Rechtsbündig entpackte Zahl mit n Dezimalstellen, links ggf. mit Nullen aufgefüllt. L Linksbündige alphanumerische Zeichenfolge, rechts ggf. mit Nullen aufgefüllt. A Rechtsbündige alphanumerische Zeichenfolge, links ggf. mit Spaces aufgefüllt. T Linksbündige alphanumerische Zeichenfolge (Text) rechts ggf. mit Spaces aufgefüllt. Bei der Definition des Überbegriffs eines Compound-Keys muß T angegeben werden. Die Unterfelder des Compound-Keys enthalten die ihrem Inhalt entsprechende Darstellung. Hat n den Wert '0' oder 'Space', so gilt das Feld als ganzzahlig definiert. Es können maximal 15 Dezimalstellen angegeben werden. Für die Zahlen '10' bis '15' ist 'A' bis 'F' einzutragen (sedezimal).
18 - 24	reserviert	Enthält Spaces
25 - 31	Wiederholfeldgruppen (OCCURS)-(starre Wiederholfelder)	
		Wiederholfelder können unterteilt werden und können an beliebigen Stellen im Satz (V-Format) oder Abschnitt (MV-Format stehen.
25 - 28	Anzahl Wiederholungen	Angabe wie oft sich das Feld wiederholt.
29 - 31	Längenangabe	Beim Wiederholfeld wird die Feldlänge, bei einer Feldgruppe wird die Summe der Feldlängen angegeben (max. 255).

Stellen	Bezeichnung	Bedeutung
32 - 35	SF1	
32 - 34	Kurzname	Eintrag eines 3 Bytes langen Kurznamens (Buchstaben und/oder Ziffern). Die Verweisdatei wird mit dem Dienstprogramm CISLADF aufgebaut. Der Inhalt des gekennzeichneten Feldes des Hauptdateisatzes wird zum Ordnungsbegriff (Primärindex) des Verweisdateisatzes. Der Ordnungsbegriff des Hauptdateisatzes bei ISAM, die Wiedergewinnungsadresse bei SAM oder die Block- und Satznummer bei PAM wird zum Satzinhalt des Verweisdateisatzes. Bei der Suche nach dem Feldinhalt greift CIS direkt auf die Verweisdatei zu und erhält somit z.B. bei ISAM die Ordnungsbegriffe jener Hauptdateisätze, in denen der gesuchte Feldinhalt vorkommt. Mit den über die Verweisdatei erhaltenen Ordnungsbegriffen greift CIS direkt auf die Hauptdateisätze zu. Für den Benutzer ist der Kurzname in der Anwendung ohne Bedeutung. Er legt damit nur den Suchpfad fest. (vgl. Manual-3: SUCHE-Kommando). Falls kein Kurzname eingetragen wird und das Feld nicht der Ordnungsbegriff ist, wird mit dem SUCHE-Kommando nach dem beschriebenen Feld sequentiell gesucht. Bei der Definition des Überbegriffs eines Compound-Keys muß ein Kurzname angegeben werden.
35 - 35	Sonderfunktion	Ein Eintrag ist nur zulässig, wenn die Stellen 32 - 34 ausgefüllt sind. D - DUPKEY wird bei Invertierung geprüft. G - Groß-/Kleinschreibung: Es wird nur Großschreibung stichwort-invertiert. K - Groß-/Kleinschreibung: Alles wird nur als Großschreibung invertiert. S - Das Feld enthält einen freien Text, der invertiert werden soll. Das A-Segment muß in diesem Fall eine Stichworttrennertabelle enthalten: EK=...E oder TK=...E (Vgl. Steuerungsparameter im A-Segment - Seite 86) Das Feld muß mit Feldbedeutung T definiert sein.

Stellen	Bezeichnung		Bedeutung
			Beispiele: Feldinhalt: MUELLER;KLAUS;MUENCHEN 1) EK=;+#E (Angabe im A-Segment) Mit dem Feldinhalt führt CIS drei Invertierungen durch: 1. MUELLER 2. KLAUS 3. MUENCHEN 2) TK=;+#E (Angabe im A-Segment) Mit dem Feldinhalt führt CIS eine Invertierung durch: KLAUS Falls Text aus einem nicht invertierten Begriff gesucht werden soll, kann die Stringsuche von CIS verwendet werden. Der Eintrag S ist in diesem Fall nicht notwendig.
		0 -	Ein Feldinhalt mit dem Wert 'Null' (bei Feldbedeutung P/H/B/R) oder 'Space' (bei Feldbedeutung T) wird nicht invertiert. Ohne diesen Eintrag werden alle Feldinhalte invertiert, unabhängig von ihrem Inhalt. Beispiele: 1) Inhalte des binären Feldes NUMMER: 4711 0000 0007 5711 0000 Mit diesen 5 Feldinhalten führt CIS 3 Invertierungen durch. 2) Inhalte des Text-Feldes NAME: MUELLER FRANKE BERGER Mit diesen 4 Feldinhalten führt CIS 3 Invertierungen durch.
36 - 39	reserviert		Enthält Spaces.
40 - 40	Wiederholabschnitt	W -	Alle Felder, die zu einem Wiederholabschnitt gehören (MV-Format) werden in dieser Spalte mit W gekennzeichnet. Der 1. Abschnitt darf kein Wiederholabschnitt sein.
		'Space'-	Das Feld gehört zu einem Abschnitt, der nicht wiederholt wird (bei MV-Format). Bei einem Satz mit V-Format darf hier nichts eingetragen werden.
41 - 44	reserviert		Enthält Spaces.

2.4.3 E-Segment der Satzbeschreibung

Stellen	Bezeichnung	Bedeutung
45 - 45	Segment	Enthält E
46 - 60	Feldbezeichnung	Die Feldbezeichnung kann maximal 15 Bytes lang sein. Buchstaben und/oder Ziffern, aber keine Spaces sind erlaubt. Die Feldbezeichnung muß innerhalb der Datenbeschreibung eindeutig sein.

Compound-Key: Bei der Definition des Überbegriffes eines Compound-Keys muß der Feldname angegeben werden. Bei allen Unterfeldern des Compound-Keys gibt es kein E-Segment.

Achtung: Bei Rechenfeldern dürfen die Zeichen + - * / : nur verwendet werden, wenn diese Feldnamen beim CIS-Kommando in Hochkomma eingeschlossen werden. Es ist nicht sinnvoll, bei Feldbezeichnungen Sonderzeichen wie = , # usw. zu verwenden, da diese bei CIS-Kommandos, z.B. SUCHEN, als Vergleichsoperatoren interpretiert werden.

2.4.4 Beispiele für Satzbeschreibungen

Beispiel 1: Satzbeschreibung bei Verwendung des V-Formats.

Das A-Segment der Datenbeschreibung enthält den Namen der Datenbank PERSB1 und folgende Parameter:

DSS=J	Suchen ist erlaubt.
DSA=J	Ausgabe ist erlaubt.
DSU=J	Update ist erlaubt.
MAXDES=35	Diese DB kann invertierte Felder zum Aufbau einer VD enthalten. Der Inhalt der invertierten Felder geht mit einer Länge von 35 Bytes in den ISAM-Schlüssel der Verweisdatei ein.
HD=HD.PERSB1	Die Hauptdatei HD.PERSB1 wird verdeckt (implizit) eröffnet.
VD=VD.PERSB1	Die Verweisdatei VD.PERSB1 wird implizit eröffnet.

Die Felder GESCHLECHT, NAME, DIENSTSTELLE und PKZ sind invertiert, unter SF1 sind die entsprechenden Segmentnamen der VD eingetragen.

```
     ----+----1----+----2----+----3----+----4----+----5----+---
 1.00  APERSB1LOGADR=J,MAXDES=35,DSS=J,DSA=J,DSU=J,
 2.00  A      HD=HD.PERSB1,VD=VD.PERSB1
 3.00  D       5  3O B                          EOB
 4.00  D       8  1  T              001         EGESCHLECHT
 5.00  D       9 35  T              002         ENAME
 6.00  D      44 35  T                          EGEBNAME
 7.00  D      79  3  T              003         EDIENSTSTELLE
 8.00  D      82 10  T                          EDATUM
 9.00  D      92 11U R              004         EPKZ
10.00  D      92  6U R                          EGEBDAT
11.00  D     103  4  P                          EGEHALT
12.00  D     107255K T          1   005G        EBEMERKUNG1
13.00  D     107255K T          2   005G        EBEMERKUNG2
14.00  D     107255K T          3   005G        EBEMERKUNG3
```

Das Beispiel ist in spaltengerechter Schreibweise im EDT-F Modus mit Spalten- und Zeilenangabe dargestellt.

Das folgende Bild zeigt Beispiel 1 in Form des mit der LSTD-Anweisung (vgl. Seite 56) erzeugten Ausdrucks. Für alle folgenden Beispiele dieses Manuals wird diese Darstellungsform verwendet.

SEGM	KOPFBESCHREIBUNG	
	NAME	PARAMETER IM A-SEGMENT
AA	PERSB1	LOGADR=J,MAXDES=35,DSS=J,DSA=J,DSU=J, HD=HD.PERSB1,VD=VD.PERSB1

FELDBESCHREIBUNG

SEGM	QUELL-DEFINITION			LOG	DARST	TV	ZIEL-DEFINITION			SONDER FUNKTION		WA	SEGM	FELDBEZEICHNUNG TRANSFORMATIONS-ERGAENZUNG
	AA	ADR	LNG				AA	ADR	LNG	SF1	SF2			
3	4	8	12	15		18	21	25	29	32	36	0	5	46 (41-44 FREI)
D		5	3	O	B								E	OB
D		8	1		T					001			E	GESCHLECHT
D		9	35		T					002			E	NAME
D		44	35		T								E	GEBNAME
D		79	3		T					003			E	DIENSTSTELLE
D		82	10		T								E	DATUM
D		92	11	U	R					004			E	PKZ
D		92	6	U	R								E	GEBDAT
D		103	4		P								E	GEHALT

E N D E

Die Spaltenangaben sind nur schematisch dargestellt. Die angegebenen Zahlen in der Überschrift der Feldbeschreibung beschreiben den Beginn der Felder.

Die Darstellung besteht aus zwei Teilen:

KOPFBESCHREIBUNG — Enthält das A-Segment.

FELDBESCHREIBUNG — Enthält das D- und E-Segment.
Die Spalten 1 - 2 sind nicht dargestellt, da sie stets frei bleiben.
Die Spalte 40 ist nur mit 0 beschrieben.
Die Spalten 41 - 44 sind nicht dargestellt, da sie stets frei bleiben.
Die Spalte 45 ist nur mit 5 beschrieben.

Erläuterungen zur Darstellung:

AA	Abschnittsart	SEGM	Segment
ADR	relative Adresse	SF1	Sonderfunktion 1
DARST	Darstellung	SF2	Sonderfunktion 2 (derzeit nicht ausgewertet)
LNG	Feldlänge	TV	Transformationsvorschrift
LOG	Logische Feldbedeutung	WA	Wiederholabschnitt

Beispiel 2: Feld- und funktionsbezogener Datenschutz

SEGM	KOPFBESCHREIBUNG	
	NAME	PARAMETER IM A-SEGMENT
AA	PERSB2	DSS=J,DSA=J,LOGADR=J,MAXDES=11,HD=HD.PERSB1,VD=VD.PERSB1

FELDBESCHREIBUNG

SEGM	QUELL-DEFINITION			LOG	DARST	TV	ZIEL-DEFINITION			SONDER FUNKTION		WA	SEGM	FELDBEZEICHNUNG / TRANSFORMATIONS-ERGAENZUNG
	AA	ADR	LNG				AA	ADR	LNG	SF1	SF2			
3	4	8	12	15		18	21	25	29	32	36	0	5	46 (41-44 FREI)
DD		5	3	O	B								EE	OB
DD		8	1		T					001			EE	GESCHLECHT
DD		79	3		T					003			EE	DIENSTSTELLE
DD		82	10		T								EE	DATUM
DD		92	11	U	R					004			EE	PKZ
DD		92	6	U	R								EE	GEBDAT
D		103	4		P								E	GEHALT

E N D E

Die Satzbeschreibung PERSB2 bezieht sich genauso wie die Satzbeschreibung PERSB1 in Beispiel 1 auf die Hauptdatei HD.PERSB1 und auf die Verweisdatei VD.PERSB1.

Es sind nur die Zugriffsarten Suchen (DSS=J) und Ausgeben (DSA=J) erlaubt, Update ist nicht zugelassen.

Die Felder NAME und GEBNAME fehlen, somit kann auf die Felder NAME und GEBNAME nicht zugegriffen werden.

Beispiel 3: A-Segment mit FILE- und Ausgabe-Parameter

SEGM	KOPFBESCHREIBUNG	
	NAME	PARAMETER IM A-SEGMENT
AAA	PERSB2	DSS=J,DSA=J,LOGADR=J,MAXDES=11,HD=HD.PERSB1,VD=VD.PERSB1 BLKSIZE=(STD,3),SPACE=(9,9),RECFORM=V ZEILEN=10/1,FORM=48

FELDBESCHREIBUNG

SEGM	QUELL-DEFINITION			LOG	DARST	TV	ZIEL-DEFINITION			SONDER FUNKTION		WA	SEGM	FELDBEZEICHNUNG TRANSFORMATIONS-ERGAENZUNG
	AA	ADR	LNG				AA	ADR	LNG	SF1	SF2			
3	4	8	12	15		18	21	25	29	32	36	0	5	46 (41-44 FREI)
DDDDDDDD		5 8 79 82 92 92 103	3 1 3 10 11 6 4	OUU	BTTTRRP					001 003 004			EEEEEEEE	OB GESCHLECHT DIENSTSTELLE DATUM PKZ GEBDAT GEHALT

E N D E

Beispiel 4: Satzbeschreibung bei Verwendung des MV-Formats

SEGM	KOPFBESCHREIBUNG												
	NAME	PARAMETER IM A-SEGMENT											
AA	KUNDEN	LOGADR=J,MAXDES=34,DSS=J,DSA=J,DSU=J,HD=HD.KUNDEN VD=VD.KUNDEN,EK= ,-.E											
	FELDBESCHREIBUNG												
SEGM	QUELL-DEFINITION			LOG	DARST	TV	ZIEL-DEFINITION			SONDER FUNKTION		WA	SEGM	FELDBEZEICHNUNG TRANSFORMATIONS-ERGAENZUNG
	AA	ADR	LNG				AA	ADR	LNG	SF1	SF2			
3	4	8	12	15		18	21	25	29	32	36	0	5	46 (41-44 FREI)
D	PERS	9	3	O	B								E	EDV-NR
D	PERS	12	30		T					001S			E	NAME
D	PERS	42	4		T					002			E	PLZ
D	PERS	46	24		T					003			E	ORT
D	PERS	70	30		T					004			E	STRASSE
D	PERS	100	5	W	T								E	WF
D	WARE	9	12		T							W	E	ARTIKEL
D	WARE	21	9		2R							W	E	PREIS
D	WARE	30	34	W	T					005		W	E	BEZEICHNUNG
	E N D E													

Beispiel 5: Der Compound-Key bei Verwendung des V-Formats

SEGM	KOPFBESCHREIBUNG	
	NAME	PARAMETER IM A-SEGMENT
AA	PERSB2	DSS=J,DSA=J,HD=HD.PERSB1,VD=VD.PERSB1,LOGADR=J, MAXDES=11

FELDBESCHREIBUNG

SEGM	QUELL-DEFINITION			LOG	DARST	TV	ZIEL-DEFINITION			SONDER FUNKTION		WA	SEGM	FELDBEZEICHNUNG TRANSFORMATIONS-ERGAENZUNG
	AA	ADR	LNG				AA	ADR	LNG	SF1	SF2			
3	4	8	12	15		18	21	25	29	32	36	0	5	46 (41-44 FREI)
DD			9	Z	T					001			E	COMPOUND-KEY
D		8	1	Z	T									
D		22	5	Z	R									
D		9	3	Z	T									
D		5	3	O	B								E	ORDNUNGSBEGRIFF
D		8	1	U	T					002			E	GESCHLECHT
D		9	3	U	T					003			E	DIENSTSTELLE
D		12	10		T								E	DATUM
D		22	11	U	R					004			E	PKZ
D		27	6	U	R								E	GEBDAT
D		33	4		P								E	GEHALT

E N D E

Der invertierte Compound-Key mit der Feldbezeichnung 'COMPOUND-KEY' besteht aus den drei Unterfeldern (Gesamtlänge = 9):

'GESCHLECHT'
5 Bytes von 'PKZ' und
'DIENSTSTELLE'.

Alle D-Segmente des Compound-Keys müssen unmittelbar aufeinanderfolgen. Der Compound-Key wurde in diesem Beispiel willkürlich als erstes D-Segment definiert.

Kurznamen (Spalte 32 - 35), Spalte 45 und Feldbezeichnungen (Spalte 46 - 60) dürfen für die Unterfelder nicht angegeben werden. Die Darstellung des Feldtyps (Spalten 16 und 17) muß der Darstellung der Quellfelder entsprechen.

Alle Quellfelder müssen in Spalte 15 ein 'U' enthalten, da sie durch den Compound-Key 'redefiniert' werden.

Beispiel 6: Der Compound-Key bei Verwendung des MV-Formats

Die folgende Satzbeschreibung COMKEY bezieht sich auf die in Beispiel 4 beschriebene Hauptdatei HD.KUNDEN.

SEGM	KOPFBESCHREIBUNG
	NAME: COMKEY — PARAMETER IM A-SEGMENT
AA	LOGADR=J,MAXDES=34,DSS=J,DSA=J,DSU=J,HD=HD.KUNDEN VD=VD.KUNDEN.COMPOUNDKEY,EK= ,-.E

FELDBESCHREIBUNG

SEGM	QUELL-DEFINITION AA	ADR	LNG	LOG DAR ST	TV	ZIEL-DEFINITION AA	ADR	LNG	SONDER FUNKTION SF1	SF2	WA	SEGM	FELDBEZEICHNUNG / TRANSFORMATIONS-ERGAENZUNG
3	4	8	12	15	18	21	25	29	32	36	0	5	46 (41-44 FREI)
D	PERS	9	3	O B								E	EDV-NR
D	PERS	12	30	T					001S			E	NAME
D	PERS	42	4	T					002			E	PLZ
D	PERS	46	24	T					003			E	ORT
D	PERS	70	30	T					004			E	STRASSE
D	PERS	100	5	W T								E	WF
D	WARE	9	12	T							W	E	ARTIKEL
D	WARE	21	9	2R							W	E	PREIS
D	WARE	30	34	U T					005		W	E	COMP-BEZEICHNG
D	WARE	30	34	W T					005		W	E	BEZEICHNUNG
D	PERS		34	Z T					006			E	CP1
D	PERS	42	4	Z T									
D	PERS	12	30	Z T									
D	WARE		15	Z T					007		W	E	CP2
D	WARE	9	3	Z T							W		
D	WARE	21	9	Z R							W		
D	WARE	30	3	Z T							W		

E N D E

Der Satz enthält 2 Compound-Keys

Name des Compound-Keys	Zusammensetzung
CP1	PLZ (in voller Länge) + Name (in voller Länge)
CP2	Die ersten 3 Bytes von ARTIKEL + PREIS in voller Länge + die ersten 3 Bytes von COMP-BEZEICHNUNG

Im Unterschied zu Beispiel 5 muß beim Compund-Key und dessen Unterfeldern der Abschnittsname der Quellfelder angegeben werden. Ein Compound-Key kann sich stets nur auf Quellfelder aus einem Abschnitt beziehen.

Der Update des Compound-Keys erfolgt über die Quellfelder, die im Compund-Key verwendet werden.

2.5 Transformationsbeschreibung

Die Transformationsbeschreibung ermöglicht es aus Sätzen einer vorgegebenen Struktur (Quellsätze) Sätze mit einer anderen Struktur (Zielsätze) zu erstellen. Sie beschreibt die Felder des Quellsatzes (Quelldefinition), die in den Zielssatz (Zieldefinition) übertragen werden. Eine Transformationsbeschreibung ist also immer dann notwendig, wenn sich Quelle und Ziel im Aufbau unterscheiden.

Anwendungen:

- Beim Laden einer Datenbank liegt der Quellsatz nicht in dem Format vor, in dem er in die Datenbank eingespeichert werden soll (vgl. Beispiel 1 - Seite 116).

- Aus einer Datenbank soll ein Extrakt in eine Datei oder in eine andere Datenbank geschrieben werden (vgl. Beispiel 2 - Seite 117).

- Der Anwender erwartet an einer Programmschnittstelle die Datenbanksätze in einem bestimmten Format, das nicht dem der Datenbank entspricht (vgl. Beispiel 3 - Seite 118).

2.5.1 A-Segment der Transformationsbeschreibung

Alle Parameter des A-Segments entsprechen denen der Satzbeschreibung (vgl. A-Segment der Satzbeschreibung - Seite 83), Ausgabeparameter werden ignoriert.

Die Auswertung der Parameter hängt von der Einsatzumgebung der Transformationsbeschreibung ab.

Unter CISLADF: Werden HD und VD gemeinsam geladen, werden nur die Parameter der Satzbeschreibung berücksichtigt.

Unter CIS: Werden die Parameter HD, DSA und EOC berücksichtigt.

2.5.2 D-Segment der Transformationsbeschreibung

Die Quelldefinition des D-Segments beschreibt die einzelnen Felder des Quellsatzes, die Zieldefinition die Felder des Zielsatzes sowie die Art der Übertragung in den Zielsatz.

Stellen	Bezeichnung	Bedeutung
1 - 2	reserviert	Enthält Spaces
3 - 3	Segment	Enthält D
4 - 14	Quelldefinition	Die Quelldefinition beschreibt die Felder des Sendesatzes. Feldbeschreibungen der Quelldefinition werden nicht ausgefüllt, enthalten also Spaces, wenn der zu übertragende Wert nicht aus dem Eingabesatz stammt oder keine Konstante ist.
4 - 7	Abschnittsname	Bei Quellsätzen mit V-Format bleibt dieses Feld unberücksichtigt. Bei Sätzen mit MV-Format wird der Abschnittsname vierstellig eingetragen.
8 - 11	Adresse	Wenn das Zielfeld aus dem Quellsatz versorgt werden soll, wird die relative Adresse des Feldes innerhalb eines Abschnittes (MV-Format) bzw. innerhalb eines Satzes (V-Format) rechtsbündig eingetragen. Führende Nullen können entfallen. Das erste Feld beginnt beim V-Format bei Adresse 5, beim MV-Format bei Adresse 9 (auch bei RECFORM=F).
	oder	
10 - 11	Konstantennummer	Wenn das Zielfeld mit einer Konstanten versorgt werden soll, wird die Nummer der Konstanten rechtsbündig eingetragen. (vgl. F-Segment - Seite 115). Eine führende Null kann entfallen. Die Spalten 8 und 9 bleiben leer.
12 - 14	Länge	Rechtsbündiger Eintrag der Länge des beschriebenen Feldes; führende Nullen können entfallen. Bei einem variabel langen Feld wird die maximale Länge eingetragen (vgl. Satzbeschreibung Feldbedeutung V - Seite 95). Die maximale Länge ist 255 Bytes. Der Längeneintrag entfällt, wenn in Spalte 15 die logische Feldbedeutung 'C' eingetragen wird. CIS ignoriert ein Feld wenn es kürzer als die angegebene Länge (z.B. letztes Feld) und nicht variabel lang ist (Feldbedeutung ungleich V). Bei der Ausgabe (z.B. ZEIGEN) stehen Blanks.

Stellen	Bezeichnung	Bedeutung	
15 - 17	Feldbedeutung		
15 - 15	Logische Feldbedeutung	C:	Das Zielfeld wird aus dem Konstantenbereich versorgt. Eintrag der Konstantennummer in Spalte 10 - 11. (vgl. F-Segment - Seite 115).
		W:	Das Quellfeld ist ein Wiederholfeld.
		V:	Das Quellfeld ist ein variabel langes Feld.
		'Space':	Das Zielfeld wird aus dem Quellbereich versorgt, es handelt sich um kein Wiederholfeld oder variabel langes Feld.
16 - 17	reserviert	Enthält Spaces	
18 - 20	Transformationsvorschrift		
18 - 18	Dezimalstellen	'0' oder 'Space':	Ganzzahlig
		'1' bis 'F'	Die maximal Anzahl Dezimalstellen ist '15'. Für die Zahlen '10' bis '15' ist 'A' bis 'F' einzutragen (sedezimal).
		Hinweis:	Ein Eintrag ungleich '0' oder 'Space ist nur beim Auftreten des externen Formats (E) in Spalte 19 oder 20 sinnvoll, da nur externe Formate einen Dezimalpunkt führen können. CIS führt keine Dezimalpunkte, Dezimalstellen werden in der Darstellung (Eintrag Spalte 16 der Satzbeschreibung) festgelegt.
19 - 19	logische Feldbedeutung des Quellfeldes		
		P, H, B, R, T oder E	
		E =	Externes Format d.h. die Feldinhalte des **Quellsatzes** werden mit Dezimalpunkt, führenden Nullen und Vorzeichen dargestellt.
20 - 20	logische Feldbedeutung des Zielfeldes		
		P, H, B, R, T, E oder D	
		E =	Externes Format d.h. die Feldinhalte des **Zielsatzes** werden mit Dezimalpunkt, führenden Nullen und Vorzeichen dargestellt.
		D =	Datum d.h. es wird das Tagesdatum erzeugt, in Spalte 18 und 19 muß folglich ein 'Space' eingetragen werden.

Bis einschließlich CIS Version 9 galten andere Transformationsvorschriften, die noch unterstützt werden. Die folgende Tabelle stellt die bis einschließlich CIS-Version 9 und die ab CIS-Version 10 geltenden Transformationsvorschriften gegenüber.

bis einschl. CIS V9	ab CIS V10	Bedeutung
Stellen 20 - 20	Stellen 18 - 20	
1	TT	Einfache Übertragung
2	PR	Entpacken
3	RP	Packen
4	D	Tagesdatum im Format TT.MM.JJJJ
5	BR	Umwandeln von binär nach dezimal entpackt
7	nPE	Aufbereitung numerischer Felder (Feldbedeutung P/R)
	nRE	Für Bildausgabe
8	RB	Umwandeln von dezimal entpackt nach binär

Transformationsmöglichkeiten

Spalte			von (Spalte 19)	nach (Spalte 20)
18	19	20		
	P	R	Gepackte Zahl	Dezimal entpackt.
n	P	E		Externes Format mit n Dezimalstellen. Es werden der Dezimalpunkt, führende Nullen und negatives Vorzeichen dargestellt.
	H	R	Gepackte vor-zeichenlose Zahl	Dezimal entpackt, Vorzeichen wird auf 'F' gesetzt, also positiv.
n	H	E		Externes Format mit n Dezimalststellen.
	B	R	Binäre Zahl	dezimal entpackt, ohne Vorzeichen, d.h. positiv.
	R	P	Dezimal entpackte Zahl	Gepackt
	R	H		Gepackt ohne Vorzeichen.
	R	B		Binär - die Quelldefinition darf maximal 10 Byte, die Zieldefinition maximal 4 Byte lang sein.
n	R	E		Externes Format mit n Dezimalstellen.
	T	T	Textfeld	Textfeld - Damit läßt sich jeder Feldtyp 1:1 übernehmen, sowie hintereinander liegende Felder (auch mit unterschiedlicher Feldbedeutung) 1:1 in einem Transformationsschritt übertragen.
	T	K		Textfelder werden komprimiert, d.h. führende und anhängende Spaces werden nicht gespeichert. Das Textfeld muß in der Ziel-beschreibung die logische Feldbedeutung K bekommen.
	K	T		Komprimierte Felder werden durch Anhängen von Spaces in Textfelder fester Länge umgewandelt. Die Feldnummer des Quellfeldes ist in Spalte 12 - 14 anzugeben.
		D	Datum erzeugen	Tagesdatum im Format TT.MM.JJJJ Im Längenfeld (Stellen 29 - 31) muß 10 angegeben werden. Die Stellen 4 bis 19 bleiben leer.
n	E	P	Externes Format	Gepackte Zahl.
n	E	H		Gepackte Zahl ohne Vorzeichen.
n	E	B		Binäre Zahl.
n	E	R		Dezimal entpackte Zahl

Stellen	Bezeichnung	Bedeutung
21 - 31	Zieldefinition	Die Zieldefinition beschreibt das Empfangsfeld im Zielsatz. Die Felder müssen nach Abschnitten zusammenhängend definiert werden.
21 - 24	Abschnittsname	Beim V-Format bleibt das Feld leer. Bei Übertragung in einen Zielsatz mit MV-Format wird der Abschnittsname vierstellig eingetragen. Der Name wird in der Regel anderweitig weiterverwendet und sollte deshalb entsprechend gewählt werden, z.B. für Programme oder Satzbeschreibungen, die sich auf den beschriebenen Zielsatz beziehen.
25 - 28	Adresse	Die relative Adresse innerhalb eines Abschnittes (bei MV-Format) bzw. eines Satzes (bei V-Format) wird rechtsbündig eingetragen. Führende Nullen können entfallen. Das erste Feld beginnt beim V-Format bei Adresse 5, beim MV-Format bei Adresse 9.
29 - 31	Länge	Rechtsbündiger Eintrag der Feldlänge im Zielsatz. Führende Nullen können entfallen.
32 - 44	reserviert	Enthält Spaces.

2.5.3 E-Segment der Transformationsbeschreibung

Im E-Segment werden bei Bedarf Transformationsergänzungen zur Behandlung von Abschnitts- und Feldmultiplitäten eingetragen. Die Abschnittsmultiplität entspricht der Reihenfolge der wiederholten Abschnitte innerhalb eines Satzes mit MV-Format, die Feldmultiplität entspricht analog der Abschnittsmultiplität der Reihenfolge der wiederholten Felder innerhalb eines Satzes mit V-Format.

In Stelle 45 wird E als Segmentname eingetragen, falls Transformations-ergänzungen ab Stelle 46 bis 80 folgen. Angaben zu Abschnitts- und Feldmultiplitäten dürfen nicht gemischt werden.

Abschnittsmultiplität

Die Multiplitätsangaben steuern die Übertragung der Abschnitte vom Quell- in den Zielsatz (vgl. Beispiel 4 - Seite 119).

Format: A(AS=k,AM=p,AZ=m[,IN=n])

AS: Anzahl der Übertragungsschritte
k: Anzahl der Übertragungsschritte, die zur selben Abschnittsmultiplität gehören. Für jeden Übertragungsschritt ist eine Zeile einzugeben. Nur in der ersten Zeile der Übertragungsschritte darf das E-Segment (E in Spalte 45) und die Multiplitätsangaben angegeben werden (vgl. Beispiel 4 - Seite 119).

AM: Abschnittsmultiplität
p: Eine positive Zahl gibt an, ab welchem Abschnitt einer bestimmten Abschnittsart des Quellsatzes in den Zielsatz übertragen wird. Beispiel: AM=3 bedeutet, daß ab dem dritten Abschnitt übertragen wird.

Eine negative Zahl bestimmt den Abschnitt, ausgehend vom letzten Abschnitt einer bestimmten Abschnittsart, ab dem übertragen wird. D.h. die Abschnitte werden vom letzen Abschnitt und nicht wie bei einer positiven Zahl vom ersten Abschnitt ausgehend numeriert.

Beispiel: AM=-2 bedeutet die Positionierung auf den vorletzten Abschnitt einer bestimmten Abschnittsart.

p=	1	2	3	4	1	2
	ABS1	ABS1	ABS1	ABS1	ABS2	ABS2
p=	-4	-3	-2	-1	-2	-1

AZ: Anzahl der Abschnittszyklen
m: Anzahl der Abschnitte, die ab dem Aufsatzpunkt, der durch AM=p definiert ist, übertragen werden. Bei AZ=$ werden alle Abschnitte des Satzes ab dem Aufsatzpunkt übertragen.

IN: Abschnittslänge des Zielsatzes
n: Die Angabe, IN=n, ist nur dann notwendig, wenn der Abschnitt in der Zieldefinition nicht vollkommen beschrieben ist.

Feldmultiplität

Die Multiplitätsangaben steuern die Übertragung der Wiederholfelder vom Quell- in den Zielsatz (vgl. Beispiel 5 - Seite 120).

Format: F(FM=m,FZ=n[,FA=A])

FM: Feldmultiplität

m: Eine positive Zahl gibt an, ab welchem Wiederholfeld des Quellsatzes in den Zielsatz übertragen wird.

Beispiel: FM=3 bedeutet, daß ab dem dritten Wiederholfeld übertragen wird.

Eine negative Zahl bestimmt das Wiederholfeld, ausgehend vom letzten Wiederholfeld, ab dem übertragen wird. D.h. die Wiederholfelder werden vom letzen und nicht wie bei einer positiven Zahl vom ersten Wiederholfeld ausgehend numeriert.

Beispiel: FM=-2 bedeutet die Positionierung auf das vorletzte Wiederholfeld.

AZ: Anzahl der Feldzyklen

n: Anzahl der Wiederholfelder, die ab dem Aufsatzpunkt, der durch FM=m definiert ist, übertragen werden. Bei FZ=$ werden alle Wiederholfelder des Satzes ab dem Aufsatzpunkt übertragen.

FA: Übertragung Wiederholfelder als Wiederholabschnitt.

A: Bei Angabe des Parameters wird jedes Wiederholfeld des Quellsatzes als Wiederholabschnitt im Zielsatz gespeichert. Fehlt dieser Parameter werden die Felder im Zielsatz lückenlos hintereinander abgelegt.

2.5.4 F-Segment der Transformationsbeschreibung

Im F-Segment werden Konstanten definiert, die in den Zielsatz übertragen werden.

Stellen	Bezeichnung	Bedeutung
1 - 2	reserviert	Enthält Spaces
3 - 3	Segment	Enthält F
4 - 5	Konstantennummer	Die Konstanten werden durch eine zweistellige Zahl gekennzeichnet, unter der in der Quelldefinition (Spalte 10 - 11) darauf Bezug genommen wird. Die Zahl wird rechtsbündig eingetragen, eine führende Null kann entfallen.
6 - 85	Konstante	Die Konstante wird einschließlich eventueller Zwischenräume entsprechend dem Längeneintrag (Spalte 29 - 31) in den Zielsatz übertragen, Entsprechend der Transformationsvorschrift in Spalte 18 - 20 des D-Segments können aus dem Zeichenformat der Konstanten auch andere Formate generiert werden, z.B. packen einer numerischen Konstanten.

2.5.5 Beispiele für Transformationsbeschreibungen

Beispiel 1: Übertragung eines Satzes in den Datenbanksatz

Die Quelldefinition der Transformationsbeschreibung PERST1 beschreibt die Felder des Quellsatzes, die in einen Zielsatz, dessen Aufbau in der Zieldefinition beschrieben ist und dem Aufbau der Datenbank HD.PERS1 (vgl. Beispiel 1 - Seite 100) entspricht, übertragen werden.

SEGM	KOPFBESCHREIBUNG	
	NAME	PARAMETER IM A-SEGMENT
A	PERST1	LOGADR=J

FELDBESCHREIBUNG

SEGM	QUELL-DEFINITION			LOG DARST	TV	ZIEL-DEFINITION			SONDER FUNKTION		WA	SEGM	FELDBEZEICHNUNG / TRANSFORMATIONS-ERGAENZUNG
	AA	ADR	LNG			AA	ADR	LNG	SF1	SF2			
3	4	8	12	15	18	21	25	29	32	36	0	5	46 (41-44 FREI)
D		90	1		TT		8	1					
D		17	35		TT		9	35					
D		52	35		TT		44	35					
D		14	3		TT		79	3					
D					D		82	10					
D		5	6		PR		92	11					
D		91	7		RP		103	4					
					E N D E								

Hinweis: Die erste mögliche Adresse der Daten im Zielsatz ist '8', da der 3-Bytes lange Schlüssel (LOGADR=J) ab Adresse '5' eingetragen wird.

Beispiel 2: Übertragung des Extrakts eines Datenbanksatzes in einen anderen Datenbanksatz

Die Quelldefinition der Transformationsbeschreibung PERST2 beschreibt einen Extrakt aus einer Datenbank (HD.PERSB1 vgl. Beispiel 1 Seite 100), der in eine andere Datenbank bzw. Datei, beschrieben in der Zieldefinition der Transformationsbeschreibung, übertragen wird.

SEGM	KOPFBESCHREIBUNG												
	NAME	PARAMETER IM A-SEGMENT											
A	PERST2	LOGADR=N											
	FELDBESCHREIBUNG												
SEGM	QUELL-DEFINITION			LOG	DARST	TV	ZIEL-DEFINITION			SONDER FUNKTION		WA	SEGM	FELDBEZEICHNUNG TRANSFORMATIONS-ERGAENZUNG
	AA	ADR	LNG				AA	ADR	LNG	SF1	SF2			
3	4	8	12	15		18	21	25	29	32	36	0	5	46 (41-44 FREI)
D		92	11			RH		5	6					
D		79	3			TT		11	3					
D		8	70			TT		14	70					
D		8	1			TT		84	1					
D		103	4			PR		85	7					
							E N D E							

Hinweis: Das Personalkennzeichen (PKZ auf Position 92 entpackt mit der Länge 11, vgl. Beispiel 1 - Seite 100) soll, da eindeutig, als neuer Ordnungsbegriff (gepackt) verwendet werden. In der noch zu erstellenden Satzbeschreibung muß das Feld als Ordnungsbegriff definiert werden.

Beispiel 3:

Ein in der Programmiersprache FORTRAN erstelltes Anwenderprogramm soll Sätze aus der Datenbank abrufen. An der Programmschnittstelle werden deshalb Datenbanksätze in FORTRAN-kompatiblem Format erwartet. Dieses Format entspricht nicht dem der Datenbank. Mit der Transformationsbeschreibung PERST3, wird das passende Format erzeugt.

SEGM	KOPFBESCHREIBUNG												
	NAME	PARAMETER IM A-SEGMENT											
A	PERST3												
FELDBESCHREIBUNG													
SEGM	QUELL-DEFINITION			LOG	DARST	TV	ZIEL-DEFINITION			SONDER FUNKTION		WA	SEGM	FELDBEZEICHNUNG TRANSFORMATIONS-ERGAENZUNG
	AA	ADR	LNG				AA	ADR	LNG	SF1	SF2			
3	4	8	12	15		18	21	25	29	32	36	0	5	46 (41-44 FREI)
DDDD		5	87			TT		5	87					
		92	6			RB		93	4					
		98	6			RB		97	4					
		104	4			PR		101	7					
								E N D E						

```
     PROGRAM MAIR
     INTEGER KL,LZI,AK,AVM,ACM,AZI
     INTEGER PKZ(25)
     CHARACTER K*80
     CHARACTER VM*80
     CHARACTER CM*4
     CHARACTER ZI*107
     CHARACTER NAM*35
     LOGICAL*1 NW1(107)
     EQUIVALENCE (K,AK),(VM,AVM),(CM,ACM)
     EQUIVALENCE (ZI,PKZ(1))
     EQUIVALENCE (ZI,NW1(1)),(NW1(9),NAM)
     EQUIVALENCE (ZI,AZI)
     KL=15
     K= 'GET,K,DB.PERST3'
     LZI=107
1    CALL CIS (KL,AK,ACM,AVM,LZI,AZI)
2    IF (CM .EQ. 'IM00') GOTO 100
     IF (CM .EQ. 'IM01') GOTO 200
     WRITE (2,200) CM,VM
     STOP
100  WRITE (2,300) NAM,PKZ(24),PKZ(25)
     STOP
200  FORMAT (' CM=',A4,' VM=',A80)
300  FORMAT (' NAME=',A35,/,'GEB-DATUM=',16,/,'REST-PKZ=',16)
     HALT
```

Beispiel 4: Transformationsbeschreibung mit Abschnittsmultiplität und Definition von Konstanten

SEGM	KOPFBESCHREIBUNG												
	NAME	PARAMETER IM A-SEGMENT											
A	TRANSX												
FELDBESCHREIBUNG													
SEGM	QUELL-DEFINITION			LOG	DARST	TV	ZIEL-DEFINITION			SONDER FUNKTION		WA	SEGM	FELDBEZEICHNUNG TRANSFORMATIONS-ERGAENZUNG
	AA	ADR	LNG				AA	ADR	LNG	SF1	SF2			
3	4	8	12	15		18	21	25	29	32	36	0	5	46 (41-44 FREI)
DD	ABS1	9	20			TT	ABS1	9	20					
DD	ABS2	9	15			TT	ABS2	9	15				E	A(AS=2,AM=1,AZ=$)
DD		01		C		TT	ABS2	24	15					
DD	ABS3	21	12			TT	ABS3	9	12				E	A(AS=2,AM=-3,AZ=2)
DD	ABS3	9	12			TT	ABS3	21	12					
SEG	LNFRD.	KONSTANTENBESCHREIBUNG												
F	01	000000000000000												

E N D E

Die Multiplitätsangaben für die Abschnitte ABS2 und ABS3 ab Spalte 46 steuern die Übertragung vom Quell- zum Zielsatz.

Die Anzahl der Übertragungsschritte AS=2 bewirkt die Übertragung zweier Felder bzw. Konstanten, die in der Transformationsbeschreibung in zwei aufeinanderfolgenden Zeilen beschrieben sind.

In Beispiel 4 wird durch die erste Transformationsergänzung im ersten Schritt ein Feld aus dem Quellsatz und im zweiten Schritt eine Konstante (hier 15 Nullen) übertragen. Durch die Angabe AM=1 wird ab dem ersten Wiederholabschnitt des Quellsatzes übertragen, durch die Angabe AZ=$ erfolgt die Übertragung aus allen Wiederholabschnitten des Quellsatzes.

Die zweite Ergänzung überträgt zwei Felder aus dem drittletzten und vorletzten Wiederholabschnitt des Quellsatzes in je einen Wiederholabschnitt des Zielsatzes. Es entstehen somit zwei Wiederholabschnitte der Abschnittsart ABS3 pro Ausgabesatz.

Beispiel 5: Transformationsbeschreibung mit Feldmultiplität

SEGM	KOPFBESCHREIBUNG												
	NAME	PARAMETER IM A-SEGMENT											
A	TRANSY												
FELDBESCHREIBUNG													
SEGM	QUELL-DEFINITION			LOG	DARST	TV	ZIEL-DEFINITION			SONDER FUNKTION		WA	SEGM	FELDBEZEICHNUNG TRANSFORMATIONS-ERGAENZUNG
	AA	ADR	LNG				AA	ADR	LNG	SF1	SF2			
3	4	8	12	15		18	21	25	29	32	36	0	5	46 (41-44 FREI)
DD		5	8			TT	ABS1	9	8				E	F(FM=1,FZ=$,FA=A)
		13	7			TT	ABS2	9	7					
							E N D E							

Die Transformationsergänzung (Spalte 46) bewirkt die Übertragung aller Wiederholfelder des Quellsatzes in Wiederholabschitte des Zielsatzes.

Beispiel 6: Transformation eines Satzes mit V-Format in einen Satz mit MV-Format

SEGM	KOPFBESCHREIBUNG												
	NAME	PARAMETER IM A-SEGMENT											
A	PERST5	LOGADR=N											
FELDBESCHREIBUNG													
SEGM	QUELL-DEFINITION			LOG	DARST	TV	ZIEL-DEFINITION			SONDER FUNKTION		WA	SEGM	FELDBEZEICHNUNG TRANSFORMATIONS-ERGAENZUNG
	AA	ADR	LNG				AA	ADR	LNG	SF1	SF2			
3	4	8	12	15		18	21	25	29	32	36	0	5	46 (41-44 FREI)
D		5	3			TT	STMM	9	3					
D		92	11			TT	STMM	12	11					
D		8	71			TT	STMM	23	71					
D		103	4			TT	FIRM	9	4					
D		79	3			TT	FIRM	13	3					
E N D E														

Die Transformationsbeschreibung PERST5 überträgt mit dem Ladeprogramm CISLADF die Felder des vorgegebenen Satzaufbaus mit V-Format (Quelldefinition) in einen Satzaufbau mit MV-Format (Zieldefinition). Die Feldinhalte werden nicht verändert, um die bestehende VD gleichzeitig für die neue HD verwenden zu können. Invertierte Felder müssen daher nicht berücksichtigt werden. Unter diesen Voraussetzungen können mehrere Felder mit einem Übertragungsbefehl transformiert werden.

Hinweis: Da der Schlüssel nicht automatisch vergeben wird (LOGADR=N), muß der 3-Bytes lange Schlüssel von Adresse 5 des V-Satzes auf Adresse 9 des MV-Satzes übertragen werden (vgl. auch Beispiel 7 - Seite 122).

Beispiel 7: Satzbeschreibung im MV-Format

SEGM	KOPFBESCHREIBUNG												
	NAME	PARAMETER IM A-SEGMENT											
AA	PERSB3	MAXDES=35,DSS=J,DSU=J, HD=HD.PERSB3,VD=VD.PERSB3											
	FELDBESCHREIBUNG												
SEGM	QUELL-DEFINITION			LOG	DARST	TV	ZIEL-DEFINITION			SONDER FUNKTION		WA	SEGM	FELDBEZEICHNUNG TRANSFORMATIONS-ERGAENZUNG
	AA	ADR	LNG				AA	ADR	LNG	SF1	SF2			
3	4	8	12	15		18	21	25	29	32	36	0	5	46 (41-44 FREI)
DD	STMM	9	3	OUR	B								EE	OB
DD	STMM	12	11	OUR	R					PKZ			EE	PKZ
DD	STMM	12	6	OUR	R								EE	GEBDAT
DD	STMM	23	1		T					GES			EE	GESCHLECHT
DD	STMM	24	35		T					NAM			EE	NAME
DD	STMM	59	35		T								EE	GEBNAME
DD	FIRM	9	4		P								EE	GEHALT
DD	FIRM	13	3		T					STO			EE	DIENSSTELLE
	E N D E													

Beispiel 7 stellt die Satzbeschreibung für die Hauptdatei HD.PERSB3 dar, die durch die Transformationsbeschreibung PERST5 aus der Hauptdatei HD.PERSB1 erzeugte wurde.

Die relative Adresse des ersten Feldes beginnt bei 9 (MV-Format). Die Feldbeschreibung beim MV-Format ist immer abschnittsbezogen.

2.6 Bildbeschreibung

Die Bildbeschreibung dient der Definition der Ausgabeformate für Drucker und/oder Datenstationen. Sie ist bis auf nachstehende Unterschiede mit der Transformationsbeschreibung (vgl. Seite 107 bis 115) identisch:

A-Segment: Der Ausgabeparameter DSA=J muß angegeben werden.

D-Segment: Quelldefinition: Die Erklärungen zu den Einträgen in der Quelldefinition der Transformationsbeschreibung gelten uneingeschränkt.

Zieldefinition:

Spalte	Bezeichnung	Bedeutung
21 - 22	Zeilennummer	Zweistellig, eine führende Null kann entfallen. Felder, die über eine gemeinsame Zeilennummer verfügen, werden in einer Zeile ausgegeben.
23 - 24	Vorschubzeichen	Die Angabe des Vorschubzeichens ist nur bei Druckausgabe notwendig.
	4n	Vorschub um n Zeilen vor dem Drucken. Das Vorschubzeichen 40 bewirkt keinen Vorschub vor sondern um eine Zeile nach dem Drucken. Für n = 10 - 15 wird A - F eingetragen.
	0n	Vorschub um n Zeilen nach dem Drucken. Für n = 10 - 15 wird A - F eingetragen.
	Cn	Vorschub auf Kanal n des Druckerlochbandes vor dem Drucken. n muß ungleich 9 sein, 1 bewirkt den Vorschub auf den Seitenanfang. Für die Kanäle 10 - 12 wird A - C eingetragen.
	8n	Vorschub auf Kanal n des Druckerlochbandes nach dem Drucken. Für die Werte von n gelten die oben angeführten Einschränkungen.
25 - 28	Adresse	Die Adresse bezieht sich auf die Position des auszugebenden Feldes in der Ausgabezeile. Das erste Zeichen einer Zeile wird mit 9 adressiert.
29 - 31	Länge	Die Längenangabe der Zieldefinition definiert die Länge des auszugebenden Feldes in der Ausgabezeile.

E-Segment: Es gelten die Erklärungen zum E-Segment der Transformationsbeschreibung.

Ausnahmen: Der Parameter IN ist unzulässig.
Der Parameter FA=A bewirkt, daß Wiederholfelder untereinander gedruckt werden.

F-Segment: Es gelten alle Erklärungen zum F-Segment der Transformationsbeschreibung.

Beispiel einer Bildbeschreibung

Die Bildbeschreibung PERST4 definiert für Drucker und/oder Datenstation die Ausgabeformate der Daten aus der Hauptdatei PERSB1. Zusätzlich werden die Daten mit Überschriften ergänzt.

SEGM	KOPFBESCHREIBUNG	
	NAME	PARAMETER IM A-SEGMENT
A	PERST4	DSA=J,HD=HD.PERSB1

FELDBESCHREIBUNG

SEGM	QUELL-DEFINITION			LOG DAR ST	TV	ZIEL-DEFINITION			SONDER FUNKTION		WA	SEGM	FELDBEZEICHNUNG TRANSFORMATIONS-ERGAENZUNG
	AA	ADR	LNG			AA	ADR	LNG	SF1	SF2			
3	4	8	12	15	18	21	25	29	32	36	0	5	46 (41-44 FREI)
D		01		C	TT	01C1	9	18					
D					D	01C1	42	10					
D		02		C	TT	0240	9	18					
D		04		C	TT	0343	9	20					
D		92	11		TT	0343	35	11					
D		05		C	TT	0441	9	4					
D		9	35		TT	0441	23	35					
D		07		C	TT	0540	9	12					
D		44	35		TT	0540	23	35					
D		03		C	TT	0640	9	13					
D		92	6		TT	0640	23	6					

SEG	LNFRD:	KONSTANTENBESCHREIBUNG
F	01	PERSONALSTAMMDATEN
F	02	----------
F	03	GEBURTS-DATUM
F	04	PERSONAL-KENNZEICHEN
F	05	NAME
F	07	GEBURTS-NAME

E N D E

Die vorstehende Bildbeschreibung erzeugt folgendes Druckbild:

```
PERSONALSTAMMDATEN                    15.04.1994


PERSONAL-KENNZEICHEN      47110004711

NAME           WILLIBALD
GEBURTS-NAME   LUEDEMANN
GEBURTS-DATUM  080857
```

2.7 Bildschirmmasken

Bildschirmmasken (immer mit Formatsteuerung) dienen zur komfortablen Erfassung und Änderung von Daten.

Eine Bildschirmmaske bezieht sich auf die Felder einer einzigen Satzbeschreibung, die bei der Definition der Bildschirmmaske unter dem Parameter 'DEF=' angegeben wird. Es besteht also eine Verknüpfung von Maske und Satzbeschreibung, so daß bei Änderung der Satzbeschreibung stets auch die Bildschirmmaske, die sich auf diese Satzbeschreibung bezieht, neu generiert werden muß (RGENM Maskenname und anschließend GENM/PUTM).

Grundsätzlich verschieden zu den Bildschirmmasken sind die Ausgabebilder ohne Formatsteuerung (Bildbeschreibungen), die nur für die Datenausgabe verwendet werden.

2.7.1 Aufbau der Bildschirmmaske

Bildschirmmasken können

im EDT als Unterprogramm von CISGEN (vgl. Seite 126)

oder

menügesteuert mit der DEFM-Anweisung (vgl. Seite 129)

erstellt werden.

Erstellung der Bildschirmmasken mit EDT

Mit der EDT-Anweisung (vgl. Seite 41) wird in den EDT als Unterprogramm von CISGEN verzweigt. Die Definition der Bildschirmmaske besteht aus drei Teilen, die auch in dieser Reihenfolge im EDT definiert werden.

1. Parametersatz
2. Datensätze
3. Endesatz

Bei der Erzeugung des CIS-internen Formats der Maske in der DABEL (GENM-/PUTM-Anweisung) erkennt CISGEN an einem Parametersatz, daß ab dieser Zeile im EDT eine Maskendefinition vorliegt.

Die folgenden Zeilen im EDT stellen die Datensätze der Maske dar, also die Daten, die die Form der Maske auf dem Bildschirm des Anwenders festlegen.

Das Ende der Datensätze und damit auch das Ende der Maskendefinition, wird durch die Angabe des Endesatzes bestimmt.

1. Parametersatz:

```
*MASK_MASK=mmmmmm,DEF=aaaaaa[,PROG=pppppp][,CHAIN=ffffff]
      [,LINES=nn][,BYTES=bbb]
```

mmmmmm	Name der Bildschirmmaske (6 Bytes)
aaaaaa	Name der Satzbeschreibung (6 Bytes)
pppppp	Name des Plausibilitätenprüfprogramms (6 Bytes)
ffffff	Name der Folgemaske (6 Bytes)
nn	Anzahl der Zeilen des Bildschirms (2-stellig mit führenden Nullen). Bei fehlender Angabe wird 24 angenommen.
bbb	Anzahl der Bytes pro Bildschirmzeile (3-stellig mit führenden Nullen). Bei fehlender Angabe wird 80 angenommen. Kürzere Zeilen können nur erzeugt werden, wenn ein 'Ende-Zeile'-Zeichen benützt wird. Alle anderen Zeilen werden auf bbb gekürzt oder mit Spaces (X'40') auf bbb verlängert.

Der Parameter 'PROG' (Programm) ist für die Verwendung eines benutzereigenen Plausibilitätenprüfprogramms (z.B. in COBOL) vorgesehen. Die entsprechende Programmschnittstelle ist im Manual-4 unter 'Plausibilitätsprüfungen bei Bildschirmmaskeneingabe' beschrieben.

2. Datensatz:

Pro Bildschirmzeile wird ein Datensatz definiert. Folgende Zeichen/Zeichenkombinationen haben die Bedeutung:

(Anfang einer Feldbezeichnung.

) Ende einer Feldbezeichnung.
Zwischen den beiden Klammern steht die Feldbezeichnung.

. Anfangsposition: Nach dieser Position steht der Feldinhalt. Die maximale Stellenanzahl des Feldinhalts muß mit Spaces dargestellt werden.

@ Anfangsposition eines dunkelgesteuerten Feldwertes.

! Anfangsposition (s.o.) eines nicht überschreibbaren Feldinhalts.

Anfangsposition (s.o.) eines numerischen Feldinhalts.
Die Tastatur ist nur für die Ziffern und die Sonderzeichen / * + - . , freigegeben.

#E Ende einer Zeile

X'0D' Neue Zeile

X'15' Neue Zeile

X'40' Space dient für Feldinhalte als Platzhalter, ansonsten als Konstante.

Sonstige Zeichen werden als Konstanten übernommen.

Folgende Längen werden als Freiraum für den Feldtwert bei . ! und # Angaben benötigt:

Feldbedeutung	minimaler Platzbedarf für den Feldinhalt in Bytes	
A, L, T	Fl	
R	Fl + 1 Fl + 2	ohne Dezimalstellen mit Dezimalstellen
B	3 5 8 10 2 * Fl + 3	Feldlänge l = 1 Feldlänge l = 2 Feldlänge l = 3 Feldlänge l = 4 Feldlänge l > 4
P	2 * Fl 2 * Fl + 1	ohne Dezimalstellen mit Dezimalstellen
Fl = Feldlänge aus der Satzbeschreibung		

Bei numerischen und gepackten numerischen Feldern (Feldbedeutung: R und P) ist zu beachten, daß der unbedingt notwendige Platzbedarf nicht nur von der Feldlänge, sondern auch von dem eventuell notwendigen Vorzeichen und dem Dezimalpunkt, d.h. von zusätzlich zwei Schreibstellen, bestimmt wird.

3. Endesatz:

Den Abschluß der Bildschirmmaskendefinition im EDT, bildet folgender EDT-Satz:

```
*MASKEND
```

Menügesteuerte Bildschirmmaskenerstellung/-änderung

Mit Hilfe der DEFM-Anweisung wird im menügesteuerten Dialog eine Bildschirmmaske erstellt (generiert) bzw. geändert. Die prinzipiellen Ausführungen zu den im EDT erstellten Bildschirmmasken (vgl. Seite 126 bis 128) sind auch hier gültig. Es entfallen lediglich die Angaben des Parametersatzes und des Endesatzes, wie sie im EDT vom Benutzer explizit anzugeben sind.

Die DEFM-Anweisung führt zu folgendem Funktionsmenü:

```
*** FUNKTIONSMENUE *** (CG70)

    BITTE NUR EIN FELD ANKREUZEN UND NAMEN
    DER BILDSCHIRMMASKE EINGEBEN

    GENERIEREN BILDSCHIRMMASKE:
    AENDERN    BILDSCHIRMMASKE:
    ENDE                      :

    NAME BILDSCHIRMMASKE      : ------
```

Generieren der Bildschirmmaske

```
*** FUNKTIONSMENUE *** (CG70)

    BITTE NUR EIN FELD ANKREUZEN UND NAMEN
    DER BILDSCHIRMMASKE EINGEBEN

    GENERIEREN BILDSCHIRMMASKE: X
    AENDERN    BILDSCHIRMMASKE:
    ENDE                      :

    NAME BILDSCHIRMMASKE      : MASK01
```

Existiert der angegebene Name der Bildschirmmaske bereits in der DABEL, wird ausgegeben:

```
*** GENERIEREN BILDSCHIRMMASKE *** (CG71)

    ANGEGEBENE BILDSCHIRMMASKE
    EXISTIERT BEREITS IN DER DABEL

    NAME BILDSCHIRMMASKE: MASK01

    VORHANDENE MASKE UEBERSCHREIBEN ?
                   JA  :
                   NEIN:

    ABBRUCH            :
```

Existiert der Name der Bildschirmmaske noch nicht in der DABEL, so ist der Name der Satzbeschreibung und bei Bedarf der Name eines Plausibilitätsprüfprogramms (vom Benutzer geschriebenes Modul, das von CIS dynamisch nachgeladen wird (vgl. Manual-4)) anzugeben:

```
*** GENERIEREN BILDSCHIRMMASKE *** (CG72)

    BITTE NAMEN DER SATZBESCHREIBUNG UND EVTL.
    NAMEN DES PLAUSIBILITAETENPROGRAMMS EINGEBEN

    NAME BILDSCHIRMMASKE         : MASK01
    NAME SATZBESCHREIBUNG        : SATZBE
    NAME PLAUSIBILITAETENPROGRAMM: ------
    ABBRUCH                      :
```

Mit der Meldung

```
DB71 LSP-TASTE DRUECKEN, DANN MASKE EINGEBEN
```

wird der Benutzer aufgefordert die Maske einzugeben. Die Hinweise zum Format der Bildschirmmaske (vgl. Seite 132) sind dabei zu beachten.

Nach Erstellen der Bildschirmmaske kann evtl. eine Folgemaske definiert werden:

```
*** FOLGEMASKE *** (CG74)

    BITTE NAMEN DER FOLGEMASKE
    EINGEBEN/MODIFIZIEREN

    NAME FOLGEMASKE: ------
```

Werden die als Name Folgemaske vorgegebenen Bindestriche `'------'` bestätigt, wird keine Folgemaske generiert und in das Funktionsmenü (CG70) verzweigt.

Bei Eingabe des Namens für die Folgemaske ist bei Bedarf der Name eines Plausibilitätsprüfprogramms anzugeben:

```
*** FOLGEMASKE *** (CG75)

    BITTE BEI BEDARF NAMEN DES PLAUSIBILITAETEN-
    PROGRAMMS EINGEBEN/MODIFIZIEREN.

    NAME FOLGEMASKE :              MASKF1
    NAME PLAUSIBILITAETENPROGRAMM: ------
```

Ändern von Bildschirmmasken

```
*** FUNKTIONSMENUE *** (CG70)

    BITTE NUR EIN FELD ANKREUZEN UND NAMEN
    DER BILDSCHIRMMASKE EINGEBEN

    GENERIEREN BILDSCHIRMMASKE:
    AENDERN BILDSCHIRMMASKE   : X
    ENDE :

    NAME BILDSCHIRMMASKE      : MASK01
```

Wird im Funktionsmenü 'AENDERN BILDSCHIRMMASKE' angekreuzt und der Name der zu ändernden Maske angegeben, so wird der Benutzer aufgefordert, die Namen der Satzbeschreibung und/oder des Plausibilitätsprüfprogramms zu modifizieren:

```
*** AENDERN BILDSCHIRMMASKE *** (CG73)

    BITTE MIT DUE-TASTE QUITTIEREN. BEI BEDARF KOENNEN
    NACHFOLGENDE ANGABEN VORHER MODIFIZIERT WERDEN.

    NAME BILDSCHIRMMASKE         : MASK01
    NAME SATZBESCHREIBUNG        : SATZBE
    NAME PLAUSIBILITAETENPROGRAMM: ------
    ABBRUCH                      :
```

Nach beendeter Änderung können Folgemasken zur Bildschirmmaske erzeugt bzw. durch Bestätigung des vorgegebenen Maskennamens '------' im Funktionsmenü (CG74) die Zuordnung bereits vorhandener Folgemasken zur Bildschirmmaske aufgehoben werden.

Bei Angabe einer bereits vorhandenen Maske, die einer anderen Satzbeschreibung zugeordnet ist, fordert CISGEN zu folgenden Entscheidungen auf:

```
*** FOLGEMASKE *** (CG76)

    DIE ANGEGEBENE FOLGEMASKE IST SCHON IN DER
    DABEL VORHANDEN UND BASIERT AUF EINER ANDEREN
    SATZBESCHREIBUNG ALS DIE VORHERGEHENDE
    MASKE. ALLE MASKEN EINER MASKENFOLGE MUESSEN
    DERSELBEN SATZBESCHREIBUNG ZUGEORDNET
    SEIN.
    SIE HABEN JETZT FOLGENDE KORREKTURMOEGLICHKEITEN:

      A) SATZBESCHREIBUNG FUER FOLGEMASKE
         KORRIGIEREN
      B) MASKENKETTE BEENDEN (------)
      C) ANDERE/NEUE FOLGEMASKE ANGEBEN

    VORHERGEHENDE MASKE
         - NAME            :
         - SATZBESCHREIBUNG:

    FOLGEMASKE
         - NAME            :
         - SATZBESCHREIBUNG:
```

2.7.2 Hinweise zum Format der Bildschirmmaske

Die Maske wird in der gleichen Form eingegeben, wie sie später in der Anwendung abgebildet wird. Dadurch ist es möglich, schon bei Maskenerstellung den Aufbau, die Spalteneinteilung und die optische Wirkung endgültig zu beurteilen.

Die Feldbezeichnungen werden in Klammern eingeschlossen, um sie von Konstanten und Steuerzeichen unterscheiden zu können. Ein in beliebigem Abstand nach der Klammer angegebenes Positionierungszeichen (. ! oder #) bestimmt die Anfangsposition des Feldwertes. Die Abbildung des Positionierungzeichens erfolgt in der Anwendung als Space.

Bei einem zu gering zugewiesenen Platz für den Feldwert wird die erste der zur Feldbezeichnung gehörenden Klammern durch ein Fragezeichen ersetzt. Nach Korrektur muß dieses Fragezeichen wieder in eine Klammer geändert werden.

Innerhalb der Bildschirmmaske müssen alle NIL-Zeichen überschrieben werden.

Soll die Maskenzeile kürzer als eine Bildschirmzeile werden (die Maske soll z.B. auch auf Bildschirmen mit kürzeren Zeilen verwendet werden), muß die Maskenzeile mit der Zeichenfolge '#E' abgeschlossen werden. Davor stehende NIL-Zeichen müssen ebenfalls überschrieben werden.

Die Bildschirmmasken müssen nicht auf dem Bildschirmtyp generiert werden, auf dem sie ablaufen sollen. Die Masken sind vom Ablauf her beliebig zwischen den einzelnen Bildschirmen austauschbar, allerdings ist bereits beim Generieren zu beachten, daß Bildschirme unterschiedliche Anzeigekapazität, Zeilenlänge und Zeilenzahl haben können.

Beim Erstellen einer Maske (GENM) wird in CISGEN geprüft, ob die letzten zwei Zeilen, die für Auswahlmenü und Fehlermeldung benötigt werden, frei sind. Sind die beiden Zeilen belegt, wird die Fehlermeldung 'FF10' ausgegeben. Die Bildschirmmaske wird trotzdem in die DABEL geladen.

Beim Generieren und Ändern von Bildschirmmasken wird die Maske auf SYSLST protokolliert. Dabei werden die möglichen Schreibstellen mit den Zeichen '@', '!' oder '#' dargestellt.

2.7.3 Beispiel einer Bildschirmmaske

Im folgenden Beispiel ist eine Bildschirmmaske mit dazugehörender Satzbeschreibung, so wie sie im EDTF-Modus unter CISGEN erstellt wurden, dargestellt.

```
     ----+----1----+----2----+----3----+----4----+----5----+---
 1.00  ASATZBEDSS=J,DSA=J,DSU=J,LAGADR=J,
 2.00  A      MAXDES=20,HD=HD.REISEKOST,VD=VD.REISEKOST
 3.00  D       5  3O B                          EOB
 4.00  D       8 20  T             NAM          ENAME
 5.00  D      28 20  T                          EVORNAME
 6.00  D      48  6U T             DAT          EDATUM
 7.00  D      48  2U T                          EJAHR
 8.00  D      50  2U T                          EMONAT
 9.00  D      52  2U T                          ETAG
10.00  D      54  4 2P                          EREISEKOSTEN
11.00  D      58 30  T                          EEINSATZORT
12.00  D      88  1  B             XYZ          EKENNZEICHEN
13.00*MASK MASK=TESTM1,DEF=SATZBE,LINES=24,BYTES=80
14.00I-------------------------------------------------------I#E
15.00I   BILDSCHIRMMASKE; ZUM AENDERN VON HD-SAETZEN        I#E
16.00I-------------------------------------------------------I#E
17.00I                                                       I#E
18.00I    FOLGENDE FELDER SIND SCHREIBGESCHUETZT:            I#E
19.00I                                                       I#E
20.00I (KENNZEICHEN) !            (NAME) !                   I#E
21.00I                          (VORNAME) !                  I#E
22.00I                                                       I#E
23.00I-------------------------------------------------------I#E
24.00I                                                       I#E
25.00I    FOLGENDE FELDER KOENNEN GEAENDERT WERDEN:          I#E
26.00I                                                       I#E
27.00I      (DATUM)       #      <                          I#E
28.00I                     JJMMTT                            I#E
29.00I      (REISEKOSTEN) #         <                      I#E
30.00I      (EINSATZORT)  .                             <  I#E
31.00I-------------------------------------------------------I#E
32.00*MASKEND
```

Erläuterung:

Das Zeichen < in der Bildschirmmaske stellt eine Konstante dar. Sie soll dem Benutzer das Ende des Eingabefeldes markieren.

Die Feldinhalte von KENNZEICHEN, NAME und VORNAME sind nicht überschreibbar, da die Schreibposition mit '!' definiert wurde.

Das 6 Byte lange (unterdefinierte und invertierte) Feld DATUM erwartet einen numerischen Wert, da die Schreibposition mit '# 'definiert wurde. (Platzbedarf 6 Bytes).

Das 4 Byte lange, mit 2 Dezimalstellen gepackte Feld REISEKOSTEN benötigt 4*2+1, also 9 Schreibstellen und muß numerisch eingegeben werden, wobei eine Ausrichtung nach dem Dezimalpunkt stattfindet. Die Eingabewerte können somit zwischen 99999.99 und -99999.99 liegen.

2.8 Verwalten der Datenbankdefinitionen und Bildschirmmasken

Die Datenbeschreibungen werden gemeinsam mit den Bildschirmmasken in der Datenbeschreibungsdatei DABEL (Standardname) verwaltet.

Ist die Datenbeschreibungsdatei beim 1. Aufruf von CISGEN noch nicht vorhanden, so wird sie von CISGEN mit KEYLEN=6, KEYPOS=7, BLKSIZE(STD,3) und SECONDARY ALLOCATION=16 erstellt.

Soll die Datei nicht standardmäßig eingerichtet werden, so muß der Benutzer vor Aufruf des CISGEN ein entsprechendes CREATE-FILE und SET-FILE-LINK Kommando geben.

Ist der Name für die Datenbeschreibungsdatei nicht DABEL, so ist dieser vor jedem Aufruf von CIS oder CIS-Dienstprogrammen mit LINK-NAME=DB im SET-FILE-LINK Kommando zuzuweisen.

2.8.1 Datenbeschreibungen und Bildschirmmasken in die DABEL aufnehmen

Die Datenbeschreibungen und Bildschirmmasken können im EDT-Speicher (vgl. EDT-Anweisung Seite 41) in dem Format aufgebaut werden, das in den vorhergehenden Abschnitten (vgl. ab Seite 83 ff) beschrieben ist.

Dieses Quellformat wird von CISGEN in das CIS-interne Format übersetzt (GEND-/PUTD-Anweisung, vgl. Seite 43 und 62 und GENM-/PUTM-Anweisung, vgl. Seite 44 und 63) und in der Datenbeschreibungsdatei (DABEL) abgespeichert. Die generierte Datenbeschreibung bzw. Bildschirmmaske wird mit Erläuterungen auf SYSLST ausgegeben.

Bildschirmmasken können aber auch im Dialog mit der DEFM-Anweisung (vgl. Seite 129) erzeugt bzw. geändert werden.

2.8.2 Datenbeschreibungen und Bildschirmmasken aus der DABEL regenerieren

Datenbeschreibungen und Bildschirmmasken, die im CIS-internen Format in der Datenbeschreibungsdatei DABEL abgespeichert sind, lassen sich mit Hilfe der RGEND- bzw. RGENM-Anweisung (vgl. Seite 67 und 68) in den EDT-Speicher schreiben (regenerieren). Mit der EDT-Anweisung (vgl. Seite 41) wird in den entsprechenden EDT-Modus verzweigt, in dem die Änderungen vorgenommen werden können. Anschließend muß die veränderte Datenbeschreibung bzw. Maske wieder in die DABEL zurückgeschrieben werden (vgl. 2.8.1).

2.8.3 Datenbeschreibungen und Bildschirmmasken definieren, duplizieren oder löschen

Definieren:

Bildschirmmasken können im Dialog mit der DEFM-Anweisung (vgl. Seite 129) definiert werden.

Datenbeschreibungen werden im EDT aufgebaut:

In CISGEN wird mit der EDT-Anweisung (vgl. Seite 41) in den EDT verzweigt. Der Benutzer gibt im EDT die Datenbeschreibungen (und Bildschirmmasken) satzweise und formatgerecht (vgl. Abschnitte ab Seite 83 ff) ein, wobei er sich von folgenden Anweisungen unterstützen lassen kann.

ADB - Anweisung (vgl. Seite 22) A-Segment definieren.

LDB - Anweisung (vgl. Seite 51) A-, D-, E-Segment definieren.

CDB - Anweisung (vgl. Seite 23) D- und E-Segmente definieren.
IDB - Anweisung (vgl. Seite 47)
SDB - Anweisung (vgl. Seite 70)
SEG - Anweisung (vgl. Seite 71)

Duplizieren:

Datenbeschreibungen oder Bildschirmmasken werden mit der RGEND- bzw. RGENM-Anweisung (vgl. Seite 67 und 68) in den EDT-Speicher geschrieben. Mit der DSWITCH-Anweisung (vgl. Seite 40) wird die andere Datenbeschreibungsdatei zugewiesen. Die Elemente, die sich im EDT-Speicher befinden, können in die neue Datenbeschreibungsdatei geschrieben werden (vgl. GEND-/PUTD-Anweisung Seite 43 und 62 bzw. GENM-/PUTM-Anweisung Seite 44 und 63).

Sollen Bildschirmmasken kopiert werden, so müssen die zugehörigen Satzbeschreibungen (erkennbar am DEF-Parameter im Parametersatz der Maske) in der neuen Datenbeschreibungsdatei vorhanden sein.

Löschen:

Mit der DELD-Anweisung (vgl. Seite 35) lassen sich eine, mehrere oder alle Datenbeschreibungen in der aktuellen DABEL löschen. Analog können Bildschirmmasken mit der DELM-Anweisung (vgl. Seite 36) in der DABEL gelöscht werden.

Die Anzahl der gelöschten Datenbeschreibungen bzw. Bildschirmmasken wird protokolliert.

3 CISLADF

3.1 Funktionsumfang

Das Dienstprogramm CISLADF baut aus einem bereits vorhandenen Datenbestand die Hauptdatei und/oder die Verweisdatei auf.

CISLADF kann sowohl über TIAM oder Batch als auch von einem Anwenderprogramm aufgerufen werden. Die gelesenen Sätze werden von CISLADF entweder unverändert übernommen oder mit einer Transformationsbeschreibung modifiziert.

CISLADF wird ausgeliefert als:

- Modul Es kann von einem Programm aufgerufen werden.
- Phase Es wird im TIAM-Dialog oder -Batch aufgerufen.

Bedienung als Modul

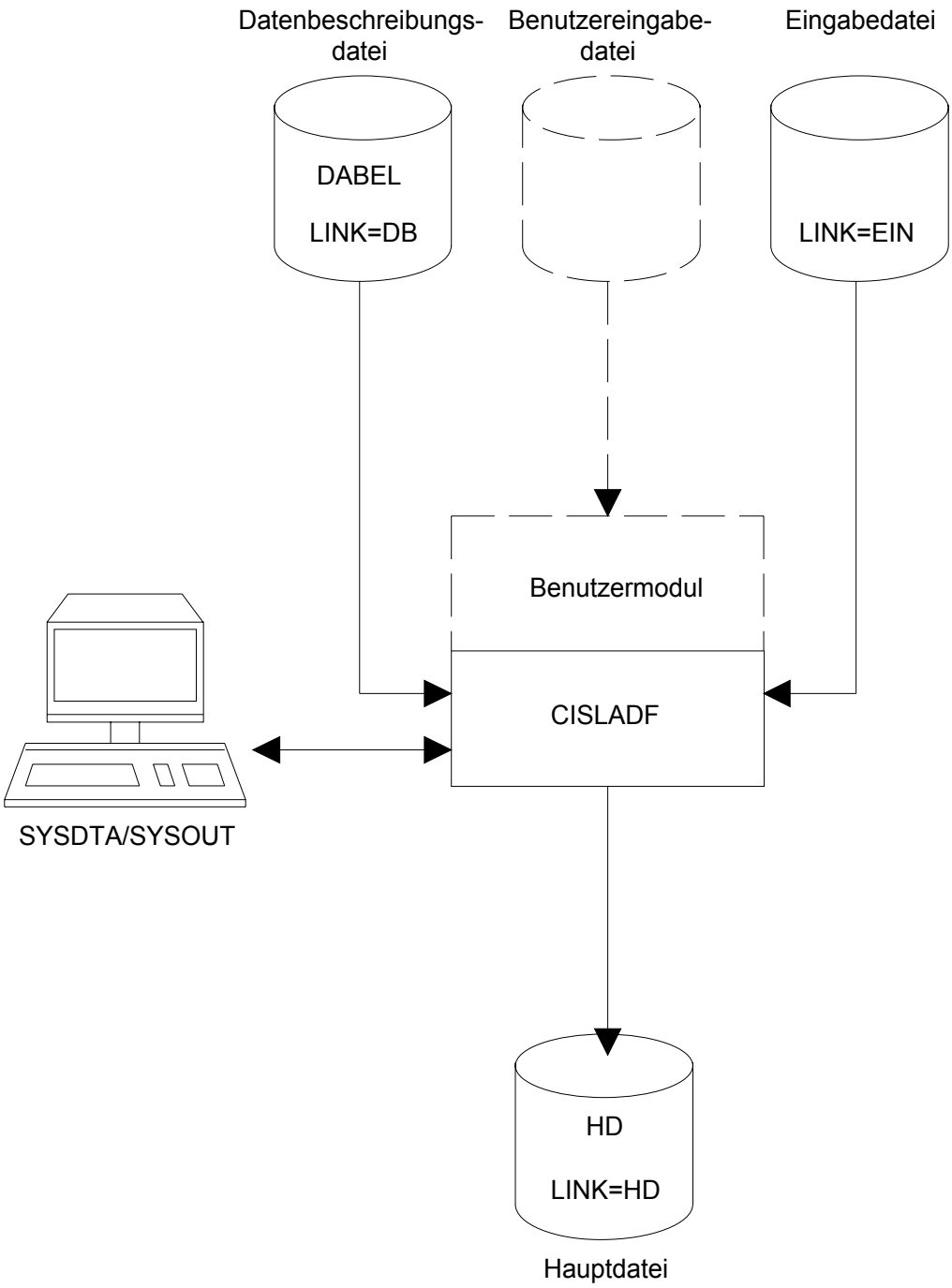
Die CISLADF-Schnittstelle ist im Manual-4 (SCHNITTSTELLEN) "CISLADF" beschrieben.

Bedienung als Phase

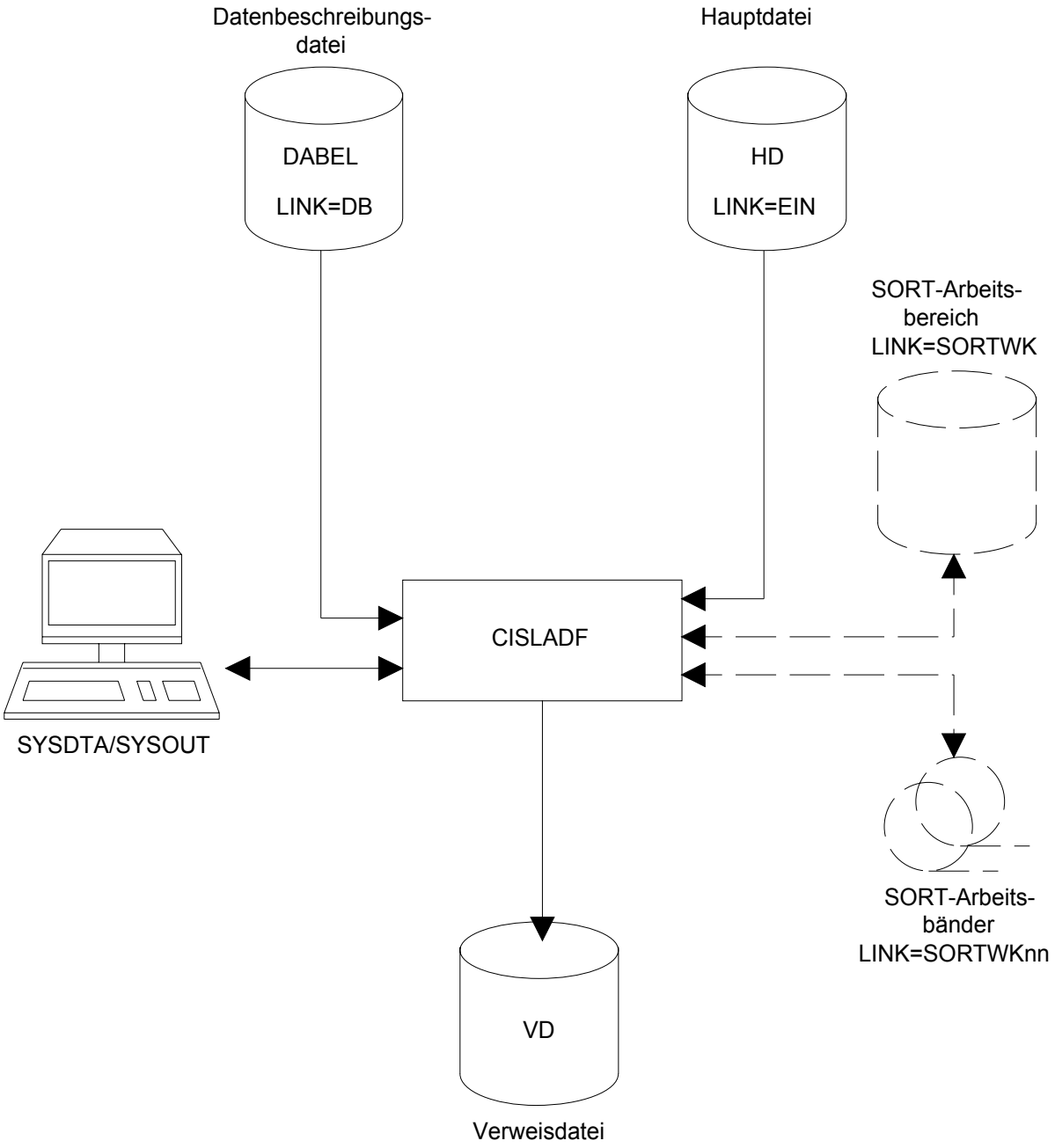
Die Eingaben werden von CISLADF über SYSDTA gelesen. die Ausgaben werden an SYSOUT übergeben.

3.2 Datenflußdiagramme

3.2.1 Datenfluß beim Laden der Hauptdatei

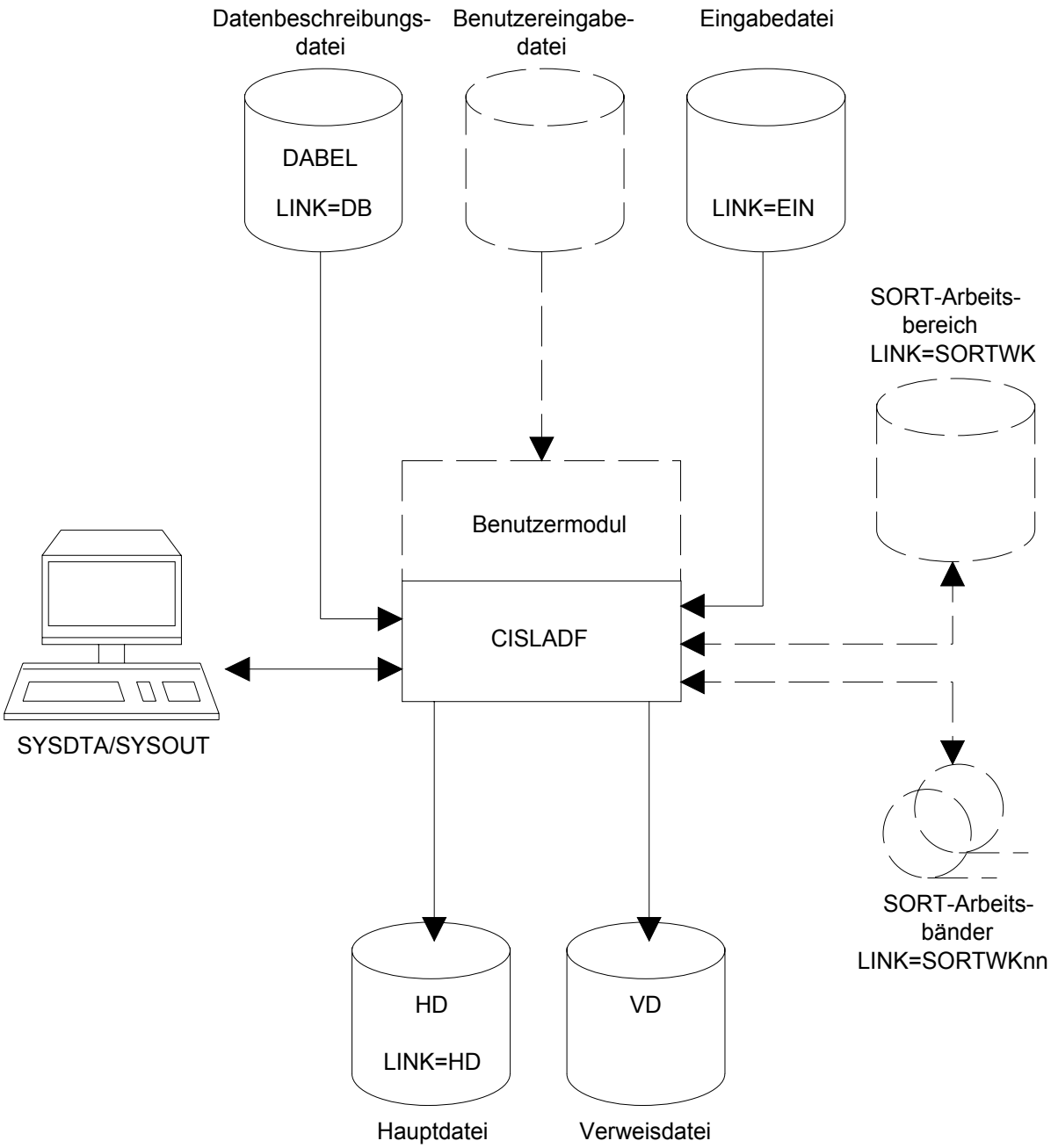


3.2.2 Datenfluß beim Laden der Verweisdatei



Aufgrund der anfallenden Datenmengen kann es sinnvoll oder notwendig sein, für den internen SORT-Ablauf Platten- und/oder Bandarbeitsbereiche zuzuweisen (siehe Beschreibung des Dientsprogrammes SORT).

3.2.3 Datenfluß beim Laden der Haupt- und Verweisdatei



Aufgrund der anfallenden Datenmengen kann es sinnvoll oder notwendig sein, für den internen SORT-Ablauf Platten- und/oder Bandarbeitsbereiche zuzuweisen (siehe Beschreibung des Dientsprogrammes SORT).

3.3 Hauptdatei laden

Das folgende Kommando baut eine CIS-Hauptdatei ohne Verweisdatei auf.

```
L,[n,]H,[EXIT=(Modul[,Bibliothek]),]DB.xxxxxx
```

L — Ladeanweisung

n — Anzahl der Sätze, die in die Hauptdatei geschrieben werden. Die Zahl n ist eine 1- bis 10-stellige Zahl. Führende Nullen können entfallen. Bei Eingabe von $ werden alle Sätze aus der Quelldatei gelesen. Fehlt die Zahl n, wird n=$ angenommen.

Bei Aufruf von CISLADF durch ein Benutzermodul (Parameter EXIT), wird n ignoriert, da vorausgesetzt wird, daß das Benutzerprogramm die Zählung übernimmt.

H — Funktionsbezeichnung für "Hauptdatei laden"

Modul — Der Benutzer übernimmt mit dem angegebenen Programm das Laden der Eingabesätze (vgl. Manual 4 "Unterprogrammanschluß")).

Bibliothek — Bei Angabe eines Bibliotheknamens wird das Benutzermodul aus der Bibliothek geladen.

xxxxxx — Satz- oder Transformationsbeschreibung

Satzbeschreibung: Die Sätze, die von CISLADF gelesen bzw. vom Benutzermodul übergeben werden, werden unverändert in den CIS-Satz übernommen. In Abhängigkeit des LOGADR-Parameters wird evtl. der Ordnungsbegriff vergeben.

Transformationsbeschreibung: Die Sätze, die von CISLADF gelesen bzw. vom Benutzermodul übergeben werden, werden entsprechend den Transformationsanweisungen übertragen. In Abhängigkeit des LOGADR-Parameters wird evtl. der Ordnungsbegriff vergeben.

Ist gleichzeitig "EXIT=..." angegeben, wird zuerst dieser User-Exit ausgeführt und anschließend die vom Unterprogramm zur Verfügung gestellten CIS-Sätze transformiert.

CISLADF liest aus einer Eingabedatei, die mit LINK-NAME=EIN zugewiesen wird. Diese Quelldatei kann eine SAM- oder ISAM-Datei mit variabler oder fester Satzlänge sein.

Wird mit einem Benutzermodul gearbeitet (Parameter EXIT im Kommando), so werden die Eingabesätze vom Benutzermodul gelesen.

Die Ausgabedatei ist immer eine ISAM-Datei und wird mit LINK-NAME=HD zugewiesen. Für die KEYLEN- und KEYPOS-Angabe im SET-FILE-LINK Kommando ist folgendes zu beachten:

Schlüssellänge: KEYLEN ist immer gleich der Länge des Ordnungsbegriffs in der Satzbeschreibung.

Schlüsselposition: Die KEYPOS-Angabe ist vom Satzformat abhängig:

V-Format und RECFORM=V	KEYPOS ist gleich der Adresse des Ordnungsbegriffs in der Satzbeschreibung.
V-Format und RECFORM=F	KEYPOS ist die Adresse des Ordnungsbegriffs in der Satzbeschreibung vermindert um 4.
MV-Format und RECFORM=V	KEYPOS ist die Adresse des Ordnungsbegriffs in der Satzbeschreibung erhöht um 4.
MV-Format und RECFORM=F	KEYPOS ist die Adresse des Ordnungsbegriffs in der Satzbeschreibung.

Beispiel für das Laden einer Hauptdatei

Aus einer SAM-Datei soll eine Hauptdatei geladen werden. Die Hauptdatei hat V-Format. Der Ordnungsbegriff ist an Stelle 5 in der Länge 3 definiert.

```
/SET-FILE-LINK FILE-NAME=EINGABE,LINK-NAME=EIN
/SET-FILE-LINK FILE-NAME=HD.LAGER,LINK-NAME=HD,ACCESS-METHOD=ISAM
(KEY-POSITION=5,KEY-LENGTH=3)
/START-PROGRAM FROM-FILE=CISLADF
L,H,DB.LAGER1
```

Allgemeines:

- Schlüssellänge

  Bei Vergabe des Ordnungsbegriffes von CIS (LOGADR=J) ist ein 3-Byte langes Binärfeld ausreichend, um mehr als 16.000.000 Sätze zu adressieren. Der Benutzer muß sich über die Eindeutigkeit des Schlüssels keine Gedanken machen. Sollte der Ordnungsbegriff bei Vergabe durch den Anwender (LOGADR=N) länger als 4 Bytes bzw. kein Binärfeld sein, muß mit einer erheblich verminderten Performance gerechnet werden.

- Urladen/Fortschreiben der Hauptdatei

  Eine Hauptdatei wird neu aufgebaut, wenn sie vorher mit dem CREATE-FILE und SET-FILE-LINK Kommando neu eingerichtet wurde. Wird eine bereits bestehende Hauptdatei zugewiesen, so wird deren Inhalt nicht gelöscht. Die Hauptdatei wird fortgeschrieben.

- Sequentielles und nicht sequentielles Schreiben

  Bei Vergabe des Ordnungsbegriffes durch CIS (LOGADR=J) wird immer sequentiell geschrieben (PUT-Makro). Die Schlüssel werden aufsteigend in Reihenfolge der geladenen Sätze vergeben. Beim Fortschreiben einer Hauptdatei werden die Ordnungsbegriffe ab dem höchsten bereits vorhandenen Schlüssel aufsteigend vergeben.

  Bei Vergabe des Ordnungsbegriffes durch den Anwender (LOGADR=N), wird nur dann sequentiell geschrieben (PUT-Makro), falls der jeweilige Ordnungsbegriff des geladenen Satzes höher ist als der bis dahin höchste in der Hauptdatei vorhandene. Andernfalls werden die Sätze eingefügt (INSRT-Makro, d.h. ein bereits bestehender Schlüssel wird nicht überschrieben, sondern führt zu einer Fehlermeldung - doppelte Schlüssel sind nicht zulässig).

  Fazit: Sequentielles Schreiben ist erheblich schneller als Einfügen. Wird der Ordnungsbegriff selbst vergeben, sollen die Quellsätze vor dem Laden nach dem gewählten Schlüssel sortiert werden.

  LOGADR=J ist die beste Voraussetzung für ein performantes Verhalten!

3.4 Verweisdatei laden

Das folgende Kommando baut aus einer bestehenden CIS-Hauptdatei eine Verweisdatei auf.

```
L,[n,]V,[VD=Datei,][VDPAD=Pad,][VDBLK=Block,]DB.xxxxxx
```

L	Ladeanweisung
n	Anzahl der Hauptdateisätze ab Dateianfang, für die eine Verweisdatei aufgebaut werden soll. Die Zahl n ist eine 1- bis 10-stellige Zahl. Führende Nullen können entfallen. Bei Eingabe von $ werden alle Sätze aus der Hauptdatei verarbeitet. Ist die Zahl n nicht angegeben, wird n=$ angenommen.
V	Funktionsbezeichnung für das Laden der Verweisdatei.
Datei	Name der Verweisdatei. Bei fehlender Angabe wird der Name der Verweisdatei in der Satzbeschreibung erwartet. Fehlt der Name der Verweisdatei (VD-Parameter) im A-Segment der Satzbeschreibung, muß der Name der Verweisdatei eingegeben werden (Voraussetzung: EOC=J). Ist der Name der Verweisdatei in der Satzbeschreibung angegeben und "Expliziter OPEN/CLOSE" (EOC-Parameter) erlaubt, so kann hier eine andere Verweisdatei als in der Satzbeschreibung angegeben werden.
Pad	Padfaktor (Füllgrad) für die Verweisdatei Bei fehlender Angabe wird VDPAD=15 angenommen.
Block	Blockfaktor für die Verweisdatei Bei fehlender Angabe wird VDBLK=1 angenommen, d.h. die Verweisdatei wird mit BLKSIZE=(STD,1) geladen.
xxxxxx	Satzbeschreibung

Die Hauptdatei wird mit LINK-NAME=EIN zugewiesen.

Die Verweisdatei wird programmintern angelegt bzw., falls sie bereits katalogisiert ist, zugewiesen. Wird eine bereits vorhandene Verweisdatei zugewiesen, so wird sie fortgeschrieben. Soll eine Verweisdatei neu aufgebaut werden, muß vor dem Ablauf von CISLADF eine evtl. vorhandene Verweisdatei gelöscht werden.

Programmintern wird eine optimale Primär- und Sekundärzuweisung für die Verweisdatei eingesetzt. Beim Programmende wird die Sekundärzuweisung auf 3 zurückgesetzt.

Wird eine bestehende Verweisdatei fortgeschrieben, sollten die Sätze am Ende der Datei sequentiell geschrieben (PUT-Makro) werden, da sonst mit Direktzugriffen gearbeitet werden muß. Dies kann zu Performanceproblemen führen. Sollen also Felder einer Hauptdatei zusätzlich invertiert werden, ist darauf zu achten, daß die Kurznamen höher sind als die früher vergebenen.

Beispiel für das Laden einer Verweisdatei

Für eine bestehende Hauptdatei soll die Verweisdatei neu geladen werden.

```
/SET-FILE-LINK FILE-NAME=HD.PERSON,LINK-NAME=EIN,
/START-PROGRAM FROM-FILE=CISLADF
L,V,DB.PERSON
```

3.5 Hauptdatei und Verweisdatei gemeinsam laden

Das folgende Kommando baut sowohl eine Hauptdatei als auch eine Verweisdatei auf.

```
L,[n,]V,[VD=Datei,][VDPAD=Pad,][VDBLK=Block,]
        [TRANS=DB.yyyyyy,][EXIT=(Modul[,Bibliothek]),]
        DB.xxxxxx
```

L	Ladeanweisung
n	Anzahl der Sätze, die in die Hauptdatei geschrieben werden. Die Zahl n ist eine 1- bis 10-stellige Zahl. Führende Nullen können entfallen. Bei Eingabe von $ werden alle Sätze aus der Quelldatei gelesen. Fehlt die Zahl n, wird n=$ angenommen. Bei Aufruf von CISLADF durch ein Benutzermodul (Parameter EXIT), wird n ignoriert, da vorausgesetzt wird, daß das Benutzerprogramm die Zählung übernimmt.
HV	Funktionsbezeichnung für das gemeinsame Laden von Haupt- und Verweisdatei .
Datei	Name der Verweisdatei. Bei fehlender Angabe wird der Name der Verweisdatei in der Satzbeschreibung erwartet. Fehlt der Name der Verweisdatei (VD-Parameter) im A-Segment der Satzbeschreibung, muß der Name der Verweisdatei eingegeben werden (Voraussetzung: EOC=J). Ist der Name der Verweisdatei in der Satzbeschreibung angegeben und "Expliziter OPEN/CLOSE" (EOC-Parameter) erlaubt, so kann hier eine andere Verweisdatei als in der Satzbeschreibung angegeben werden.
Pad	Padfaktor (Füllgrad) für die Verweisdatei Bei fehlender Angabe wird VDPAD=15 angenommen.
Block	Blockfaktor für die Verweisdatei Bei fehlender Angabe wird VDBLK=1 angenommen, d.h. die Verweisdatei wird mit BLKSIZE=(STD,1) geladen.
yyyyyy	Transformationsbeschreibung Bei unterschiedlichem Satzaufbau zwischen Quell- und CIS-Hauptdatei werden die Sätze entsprechend der angegebenen Transformationsbeschreibung von der Quell- in die Hauptdatei übertragen. Ist gleichzeitig "EXIT=..." angegeben, wird zuerst dieser User-Exit ausgeführt und anschließend die vom Unterprogramm zur Verfügung gestellten CIS-Sätze transformiert.

`Modul` Der Benutzer übernimmt mit dem angegebenen Programm das Laden der Eingabesätze (vgl. Manual 4 "Unterprogrammanschluß").

`Bibliothek` Bei Angabe eines Bibliotheknamens wird das Benutzermodul aus der Bibliothek geladen.

`xxxxxx` Satzbeschreibung

Die Satzbeschreibung beschreibt die Datenbank, d.h. ob CIS oder der Anwender den Ordnungsbegriff vergibt, welche Felder invertiert werden und wie die Verweisdatei heißt.

Laden Hauptdatei

CISLADF liest aus einer Eingabedatei, die mit LINK-NAME=EIN zugewiesen wird. Diese Quelldatei kann eine SAM- oder ISAM-Datei mit variabler oder fester Satzlänge sein.

Wird mit einem Anwendermodul gearbeitet (Parameter EXIT im Kommando), so werden die Eingabesätze vom Anwendermodul gelesen.

Die Ausgabedatei ist immer eine ISAM-Datei und wird mit LINK-NAME=HD zugewiesen. Für die KEYLEN- und KEYPOS-Angabe im SET-FILE-LINK Kommando ist folgendes zu beachten:

Schlüssellänge KEYLEN ist immer gleich der Länge des Ordnungsbegriffs in der Satzbeschreibung.

Schlüsselposition Die KEYPOS-Angabe ist vom Satzformat abhängig:

V-Format und RECFORM=V	KEYPOS ist gleich der Adresse des Ordnungsbegriffs in der Satzbeschreibung.
V-Format und RECFORM=F	KEYPOS ist die Adresse des Ordnungsbegriffs in der Satzbeschreibung vermindert um 4.
MV-Format und RECFORM=V	KEYPOS ist die Adresse des Ordnungsbegriffs in der Satzbeschreibung erhöht um 4.
MV-Format und RECFORM=F	KEYPOS ist die Adresse des Ordnungsbegriffs in der Satzbeschreibung.

Allgemeines:

- Schlüssellänge

  Bei Vergabe des Ordnungsbegriffes durch CIS (LOGADR=J) ist ein 3-Byte langes Binärfeld ausreichend, um mehr als 16.000.000 Sätze zu adressieren. Der Benutzer muß sich über die Eindeutigkeit des Schlüssels keine Gedanken machen. Sollte der Ordnungsbegriff bei Vergabe durch den Anwender (LOGADR=N) länger als 4 Bytes bzw. kein Binärfeld sein, muß mit einer erheblich verminderten Performance gerechnet werden.

- Urladen/Fortschreiben der Hauptdatei

  Eine Hauptdatei wird neu aufgebaut, wenn sie vorher mit dem SET-FILE-LINK Kommando neu eingerichtet wurde. Wird eine bereits bestehende Hauptdatei zugewiesen, so wird deren Inhalt nicht gelöscht. Die Hauptdatei wird fortgeschrieben.

- Sequentielles und nicht sequentielles Schreiben

  Bei Vergabe des Ordnungsbegriffes durch CIS (LOGADR=J) wird immer sequentiell geschrieben (PUT-Makro). Die Schlüssel werden aufsteigend in Reihenfolge der geladenen Sätze vergeben. Beim Fortschreiben einer Hauptdatei werden die Ordnungsbegriffe ab dem höchsten bereits vorhandenen Schlüssel aufsteigend vergeben.

  Bei Vergabe des Ordnungsbegriffes durch den Anwender (LOGADR=N), wird nur dann sequentiell geschrieben (PUT-Makro), falls der jeweilige Ordnungsbegriff des geladenen Satzes höher ist als der bis dahin höchste in der Hauptdatei vorhandene. Andernfalls werden die Sätze eingefügt (INSRT-Makro, d.h. ein bereits bestehender Schlüssel wird nicht überschrieben, sondern führt zu einer Fehlermeldung - doppelte Schlüssel sind nicht zulässig).

  Fazit: Sequentielles Schreiben ist erheblich schneller als Einfügen. Wird der Ordnungsbegriff selbst vergeben, sollen die Quellsätze vor dem Laden nach dem gewählten Schlüssel sortiert werden.

  LOGADR=J ist die beste Voraussetzung für ein performantes Verhalten!

Laden Verweisdatei

Die Verweisdatei wird programmintern angelegt bzw., falls sie bereits katalogisiert ist, zugewiesen. Wird eine bereits vorhandene Verweisdatei zugewiesen, so wird sie fortgeschrieben. Soll eine Verweisdatei neu aufgebaut werden, muß vor dem Ablauf von CISLADF eine evtl. vorhandene Verweisdatei gelöscht werden.

Programmintern wird eine optimale Primär- und Sekundärzuweisung für die Verweisdatei eingesetzt. Beim Programmende wird die Sekundärzuweisung auf 3 zurückgesetzt.

Wird eine bestehende Verweisdatei fortgeschrieben, sollten die Sätze am Ende der Datei sequentiell geschrieben (PUT-Makro) werden, da sonst mit Direktzugriffen gearbeitet werden muß. Dies kann zu Performanceproblemen führen. Deshalb sollen die Kurznamen der Felder, die invertiert werden, höher sein als die früher vergebenen.

Es ist darauf zu achten, daß bei Zuweisung der Haupt- und Verweisdatei beide Dateien entweder neu aufgebaut oder fortgeschrieben werden, da andernfalls ein inkonsistenter Datenbestand erzeugt wird.

Beispiel für das gemeinsame Laden einer Haupt- und Verweisdatei

Aus einer ISAM-Datei sollen die ersten 200 Sätze in eine CIS-Datenbank geladen werden. Die Hauptdatei hat MV-Format. Der Ordnungsbegriff ist an Stelle 9 in der Länge 3 definiert.

```
/SET-FILE-LINK FILE-NAME=DABEL.XY,LINK-NAME=DB
/SET-FILE-LINK FILE-NAME=EINGABE,LINK-NAME=EIN
/CREATE-FILE FILE-NAME=HD.KUNDEN,SUPPORT=PUBLIC-DISK(SPACE=RELATIVE
(PRIMARY-ALLOCATION=30,SECONDARY-ALLOCATION=30))
/SET-FILE-LINK FILE-NAME=HD.KUNDEN,LINK-NAME=HD,ACCESS-METHOD=ISAM
(KEY-POSITION=13,KEY-LENGTH=3)
/CREATE-FILE FILE-NAME=SORTBER,SUPPORT=PUBLIC-DISK(SPACE=RELATIVE
(PRIMARY-ALLOCATION=50,SECONDARY-ALLOCATION=20))
/SET-FILE-LINK FILE-NAME=SORTBER,LINK-NAME=SORTWK
/START-PROGRAM FROM-FILE=CISLADF
L,200,HV,VDPAD=0,DB.KUNDEN
/DELETE-FILE SORTBER
```

4 CISDBH

4.1 Funktionsumfang

CISDBH, der unabhängige (=independent) CIS Data-Base-Handler, wird als fertig gebundenes Programm ausgeliefert.

CISDBH führt alle CIS-Kommandos aus. Das Benutzerprogramm verkehrt mit ihm über das Modul CISCON. Die Verbindung zwischen CISCON und CISDBH kann über P1-Eventing und Common-Memory-Pool oder über DCAM realisiert werden (vgl. Datenflußpläne - Seite 177). Im aktiven CIS-Betrieb ist der Kurzkommando-Interpreter CISKURZ mit dem Modul CISCON zusammengebunden.

CISDBH kann pro Rechner mehrmals vorhanden sein. Um sie voneinander unterscheiden zu können, erhält jeder eine 1-oder 2-Byte-Idendifikation (auch DBHID genannt).

Außerdem können mehrere Tasks für eine DBHID arbeiten (Multitask-Betrieb). Die Gruppe dieser Tasks hat eine DBHID und wird von außen über diese adressiert. Die Verteilung auf die einzelnen Tasks geschieht vollautomatisch. Im Multitask-Betrieb darf kein inlinked CISKOOR benützt werden.

Die Anpassung von CISDBH an eine spezielle Umgebung (z.B. Datensicherung = ja/nein) wird über Parameter, die mit einer ENTER-Prozedur (vgl. Seite 153) an CISVARI übergeben werden, gesteuert. CISVARI enthält alle variablen Parameter für CIS.

CISDBH benötigt folgende Dateien:

- Datenbeschreibungsdatei (DABEL)
- Hauptdateien (HD)
- Verweisdateien (VD)
- ISISOUT-Dateien bei WRITE-Kommando
- Datensicherungsdateien, wenn mit eingebundenen (inlinked) CISKOOR gearbeitet wird.

Die Kurzkommandodatei (KUKO) wird nicht von CISDBH bearbeitet.

CISDBH läuft als Transaktionsprozess, sofern dies im Join-Eintrag erlaubt ist.

CISDBH-V12 kann als Partner nur Programme bedienen, die ein CISCON-V12 benützen. Alle CISDBH-Tasks einer Gruppe müssen die gleiche Version haben.

4.2 Bedienung

CISDBH wird in einem ENTER-Prozeß (vgl. Seite 153) gestartet. Alle Anpassungen an die jeweiligen Belange werden in diesem Prozeß über Parameter gesteuert.

Eingriffe während des Betriebs können über das SEND-MESSAGE-Kommando oder über das CIS-Kommando $D vorgenommen werden.

Folgende Prozeß-Schalter werden von CISDBH benützt: (CISDBH setzt beim Start die Prozeß-Schalter 25, 26 und 27 auf OFF)

25 = ON Wenn es beim CISDBH-Lauf eine CIS-Fehlermeldung gab.

26 = ON Wenn CISDBH durch einen Fehler in CISDBH mit x'00' < IW < x'80' abgebrochen wurde.

27 = ON Wenn CISDBH durch einen Fehler in CIS abgebrochen wurde. CISDBH kann neu geladen werden (z.B. SKIP-COMMANDS Kommando in der Start-Prozedur).

4.2.1 Einrichten von CISDBH

Im ausgelieferten Programm CISDBH ist das Modul CISVARI (mit Datensicherung), Programm für die Einstellung der Parameter für den Programmablauf (vgl. Manual 4), eingebunden.

CISDBH wird mit einer ENTER-Prozedur, in der die gewünschten Parameter gesetzt werden, eingerichtet.

CISDBH kann - ohne Datensicherung
- mit eingebundenen CISKOOR, oder
- mit unabhängigen CISKOOR

arbeiten. Hier gelten dieselben Regeln wie beim eingebundenen (inlinked) CIS.

4.2.2 Start des CISDBH-Prozesses

Folgende ENTER-Prozedur muß definiert werden.

```
/LOGON
/SET-FILE-LINK LINK-NAME=DB,FILE-NAME=Name
/SET-FILE-NAME LINK-NAME=AIM,FILE-NAME=Name
/SET-FILE-LINK LINK-NAME=BIM,FILE-NAME=Name
/SET-FILE-LINK LINK-NAME=PRO,FILE-NAME=Name
/START-PROGRAM FROM-FILE=CISDBH

DBHID=i
   .
   .
   .
END

/LOGOFF
```

DABEL

Parameter sind nur anzugeben wenn mit eingebundenem CISKOOR gearbeitet wird.

Parameter für CISDBH

Der Prozess wird mit einem

- ENTER-JOB Kommando durch den Benutzer oder ein
- automatisches ENTER-JOB Kommando in CISCON (siehe ENT-JOB Parameter) gestartet. Der Name der ENTER-Datei muß den String CISDBH.i enthalten.

  i = Identifikation von CISDBH, vgl. DBHID=i - Seite 151.

  In der ENTER-JOB Routine ist i bei 1-Byte Kennung ein Byte, bei 2-Byte Kennung jedoch zwei Bytes lang.

4.2.3 Parameter für Ablaufumgebung von CISDBH

Übersicht:

Parameter	Bedeutung	Default-wert
`* . . .`	Kommentar	
`CISVARI+d={C'c...c'/ X'xx..xx'}`	Korrektur von CISVARI (variable Parameter)	
`DADM=J/Y/N`	Festlegung des 1. CISDBH-Task als Verwaltungstask	N
`DAIM=J/Y/N`	Verwendung der AIM-Datei	J
`DBHID=i`	Festlegung der Identifikation des DBH	A
`DBHPW=p`	Paßwort für Verbindungsaufbau	Spaces
`DBIM=J/Y/N`	Verwendung der BIM-Datei	J
`DCAM=J/Y/N`	Verwendung des DCAM-Anschlusses	N
`DEND=e`	Setzen der Beendigungsbedingungen	2
`DENT-JOB=Parameter`	Setzen der Parameter zum Starten weiterer CISDBH-Tasks	Spaces
`DERR=cccc,E/D/T/K`	Reaktion auf Fehlercode	
`DFROM=hhmm`	Betriebszeit: Setzen der Startzeit	0700
`DKOOR=J/I/N`	J = unabhängiger CISKOOR I = eingebundener CISKOOR N= keine Datensicherung	J
`DLOG=J/Y/N`	Verwendung Protokolldatei	N
`DMAXT=n`	Maximale Anzahl der Tasks für CISDBH	1
`DMEMPAGE=n`	Anzahl Pages des Speichers für Datenbeschreibungen	64
`DNCOMM=n`	Anzahl Partner für CISDBH	100
`DNRIO=n`	Anzahl der im Speicher verwalteten Transaktionen	50
`DNTASKS=n`	Anzahl der Tasks, die für CISDSBH automatisch gestartet werden	1

Parameter	Bedeutung	Default-wert
DNTRANS	Anzahl der Transaktionen CISDBH	200
DPASS=pppppp[,J]	Angabe eines CIS-Paßwortes	
DPOOL=n	Festlegung der Memory-Pool-Größe	16
DSTAT=J/Y/N	Ausgabe von Statistikwerten am Ende des Programms	N
DTO=hhmm	Betriebszeit: Setzen der Endezeit	1800
DTRANS=J/Y/N	Transaktionen	J
DVAIM=J/Y/N	Verwendung einer VD-AIM	N
END[_Parameter]	Ende der Parameter [Umweisung von SYSDTA]	
ENT-JOB=Parameter	Parameter zum Starten CISKOOR-Task	Spaces
SYSDTA=Name	Zuweisung von SYSDTA	

Jeder Parameter kann einen Kommentar enthalten. Er muß vom Parameter durch mindestens ein Space getrennt sein.

```
*...
```

Kommentar: Es können beliebig viele Kommentare vorhanden sein.

```
CISVARI+d{C'c...c'/X'xx..xx'}
```

Korrektur von CISVARI (variable Parameter für die Ablaufumgebung von CIS).

d	Distanz CISVARI (sedizimal anzugeben, maximal 4-stellig).
c...c	Text für CISVARI.
xx...xx	Sedezimaler Text für CISVARI (immer gerade Zahl von Zeichen).

Es können mehrere CISVARI-Parameter angegeben werden.

```
DADM=J/Y/N
```

Einrichten des Verwaltungstasks

Angabe, ob der 1. CISDBH-Task nur als Verwaltungstask laufen soll. Der Verwaltungstask verarbeitet keine Nachrichten von den angeschlossenen Partnern. Er kann über das /SEND-MESSAGE Kommando Verwaltungsaufrufe entgegennehmen und verarbeiten.

Der Verwaltungstask ist an die gesamte CIS-Umgebung angeschlossen. Sollten andere Tasks auf Grund von CIS-Fehlern beendet und wieder geladen werden, dann hält in dieser Zeit der Verwaltungstask alle Memory-Pools fest.

Bei DCAM=J/Y ist der 1. Task der DCAM-Primary. Mit diesem Parameter kann somit der Primary von allen DCAM-Übertragungsfunktionen freigehalten werden.

J/Y	1. Task ist Verwaltungstask
N	1. Task ist kein Verwaltungstask

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird N angenommen.

```
DAIM=J/Y/N
```

Datensicherung: Verwendung der After-Images.

J/Y After-Images
N Keine After-Images

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird J angenommen.

```
DBIM=J/Y/N
```

Datensicherung: Verwendung Before-Images.

J/Y Before-Images
N Keine Before-Images

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird J angenommen.

```
DBHID=i
```

Identifikation des Data-Base-Handlers festlegen.

i 1-oder 2-Byte-DBH-Kennung (Buchstaben A-Z oder Ziffern 0-1).

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird als Kennung A angenommen.

```
DBHPW=p
```

Definition eines Paßwortes für den Verbindungsaufbau. CISCON muß genau dieses Paßwort angeben.

p 1-bis 8-stelliger alphanumerischer Wert.

```
DCAM=J/Y/N/A
```

Steuern des DCAM-Anschlusses:

J/Y Kommunikation nur über DCAM.

N Kommunikation nur über P1-Eventing.

A Wie DCAM=N, aber CISDBH meldet sich bei DCAM als Anwendung CISDBHi (i = DBHID) an; kann mit /BCEND oder /BCLOSE CISDBHi beendet werden.

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so gibt es keinen DCAM-Anschluß.

Gibt es einen DCAM-Anschluß, so meldet sich der Data-Base-Handler bei DCAM mit dem Namen CISDBHi an. (i = DBH-Kennung - siehe DBHID-Parameter)

Dieser Parameter kann bis zu 32 mal angegeben werden.

```
DEND=e
```

Setzen der Beendigungsbedingung des Data-Base-Handlers.

e Beendigungsbedingung (1, 2, 3)

1 = CISDBH wird nach Abmeldung des letzten Partners beendet.

2 = CISDBH kann während der Betriebszeit (vgl. DFROM- und DTO-Parameter - Seite 159 und 163) beendet werden. Außerhalb dieser Zeit geht CISDBH automatisch auf die Beendigungsbedingung 1.

3 = CISDBH kann nicht beendet werden. Die Beendigungsbedingung muß, falls CISDBH beendet werden soll, mit

/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))],MESSAGE=DEND e

oder mit $D DEND auf 1 oder 2 gesetzt werden.

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird der Wert 2 angenommen.

```
DENT-JOB=Parameter
```

Parameter zum Starten weiterer CISDBH-Tasks (siehe DNTASKS=n).

`Parameter` Parameter für ENTER-JOB Kommando (maximal 200 Bytes).
Vgl. BS2000-Kommando /ENTER-JOB: Der Parameter wird ab dem Operanden "Dateiname" angegeben.(Es wird also /ENTER-JOB FROM-FILE= davor gesetzt.)

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so kann die Anzahl Tasks nicht mit dem CISDBH-Kommando erhöht werden, sondern nur durch manuelles Starten der zusätzlichen Tasks.

```
DERR=cccc,E/D/T/K
```

Parameter, um im Fehlerfall gezielt reagieren zu können.

`cccc` Fehlercode (Codierte Meldung von CISDBH)

E Bei Fehler cccc wird ein Dump ausgegeben. CISDBH läuft normal weiter. Bei Mehrtaskbetrieb ist nur der Task betroffen, in dem der Fehler erkannt wurde.

D Bei Fehler cccc wird ein Dump ausgegeben. CISDBH läuft normal weiter. Bei Mehrtaskbetrieb sind alle Tasks betroffen.

T Bei Fehler cccc wird ein Dump ausgegeben und CISDBH wird beendet. Bei Mehrtaskbetrieb werden alle anderen Tasks gleich behandelt.

K Bei Fehler cccc werden alle CISDBH-Tasks mit Dump beendet. Alle Partner werden beim nächsten Aufruf von CISDBH ebenfalls mit Dump beendet.

```
DFROM=hhmm
```

Betriebszeit: Setzen der Startzeit, ab der CISDBH bei Beendigungsbedingung 2 nicht beendet werden kann.

`hhmm` Zeit in der Form: Stunde, Minuten

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird 0700 angenommen.

```
DKOOR=J/I/N
```

Verwendung CISKOOR

J Independent CISKOOR
I Inlinked CISKOOR
N Keine Datensicherung

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird J angenommen.

```
DLOG=J/Y/N
```

Verwendung Protokolldatei.

J/Y Protokolldatei
N Keine Protokolldatei

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird N angenommen.

```
DMAXT=n
```

Maximale Anzahl Tasks für CISDBH. Im laufenden Betrieb kann die Anzahl Tasks jederzeit zwischen 1 und dem maximalen Wert geändert werden.

n Max. Anzahl Tasks für CISDBH

Voraussetzung: DKOOR=N oder J

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird der Wert 1 angenommen (Kein Multitask-Betrieb).

```
DMEMPAGE=n
```

Anzahl Speicherseiten für die Datenbeschreibungen. Dieser Speicher ist ein Common-Memory-Pool in dem alle Datenbeschreibungen (auch Verbundbeschreibungen) gespeichert werden.

n Anzahl Speicher-Seiten (je 4K).

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird der Wert 64 angenommen.

```
DNCOMM=n
```

Anzahl der Partner für CISDBH.

n Max. Anzahl der Partner, die CISDBH verwalten soll

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird der Wert 100 angenommen.

```
DNRIO=n
```

Anzahl der Transaktionen, die im Speicher verwaltet werden. Die übrigen werden in der PAM-Datei CISUTM.RIOFILE.CISDBHi (i =DBH-Kennung) verwaltet.

n Anzahl Transaktionen, die im Speicher verwaltet werden. Bei n = 0 werden alle Transaktionen in der PAM-Datei verwaltet.

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird der Wert 50 angenommen.

Insbesondere wird bei Gleichheit von DNTRANS (max. Anzahl von Transaktionen unter CISDBH) und DNRIO keine PAM-Datei zum Speichern der Terminalbereiche angelegt.

```
DNTASKS=n
```

Anzahl der Tasks für CISDBH nach dem Start. Wenn n > 1 muß der Parameter DENT-JOB=Parameter vorhanden sein.

n Anzahl Tasks für CISDBH (≤ DMAXT-Angabe, s.o.)

Voraussetzung: DKOOR=N oder J

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird der Wert 1 angenommen.

```
DNTRANS=n
```

Anzahl der Transaktionen CISDBH

n Max. Anzahl der Transaktionen, die CISDBH verwalten kann

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird der Wert 200 angenommen.

```
DPASS=pppppp[,J]
```

Angabe eines CIS-Paßwortes.

pppppp CIS-Paßwort (immer 6-stellig)

J pppppp ist ein ein Join-Paßwort, es wird keine Datei eröffnet.

Dieser Parameter darf höchstens 512 mal angegeben werden.

Mit jedem der angegebenen Paßworte ruft CISDBH CIS auf und eröffnet somit diese Datenbank.

Meldet sich ein Partner bei CISDBH an, wird ihm die Liste der Paßworte mitgeteilt. Aufgrund dieser Liste wird in CISCON die Verteilregel ermittelt (in CISCON muß der PARTNER-Parameter für diesen CISDBH angegeben werden. Vgl. Seite 192 und 193).

Ist der DPASS-Parameter nicht im CISDBH vorhanden, so erhält CISCON seine Verteilregel über eigene Parameter (siehe DBH-und DIST-Parameter. Vgl. Seite 187 und 188).

```
DPOOL=n
```

Festlegen der Memory-Pool-Größe.

n Größe des Memory-Pools in Vielfachem von 64 K.

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird der Wert 16 angenommen.

```
DSTAT=J/Y/N
```

Ausgabe der Statistikwerte bei Programmende.Die Statistik wird vom CIS-Team ausgewertet.

J/Y Statistik ausgeben

N Keine Statistik aufgeben

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird N angenommen.

```
DTO=hhmm
```

Betriebszeit: Setzen der Endezeit, bis zu der der CISDBH nicht beendet werden kann, wenn die Beendigungsbedingung 2 ist.

hhmm Zeit in der Form: Stunde, Minuten

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, wird 1800 angenommen.

```
DTRANS=J/Y/N
```

Angabe, ob mit Transaktionen gearbeitet wird.

J/Y Transaktionen
N Keine Transaktionen

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird J angenommen.

```
DVAIM=J/Y/N
```

Einsatz der VD-After-Images als Datensicherung

J/Y VD-After-Images
N Keine VD-After-Images

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird N angenommen.

```
END[_Parameter]
```

Ende der Parameter. Muß immer angegeben werden.

`Parameter` Umweisen von SYSDTA. Es wird intern das Kommando

`/ASSIGN-SYSDTA TO-FILE=Parameter`

abgesetzt.

```
ENT-JOB=Parameter
```

Parameter zum Starten CISKOOR-Task (automatischer ENTER).

`Parameter` Parameter für ENTER-JOB Kommando (maximal 200 Bytes).
Vgl. BS2000-Kommando /ENTER-JOB: Der Parameter wird ab dem Operanden "Dateiname" angegeben. (Es wird /ENTER-JOB FROM-FILE= davor gesetzt.)

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, wird CISKOOR nicht automatisch gestartet.

```
SYSDTA=Name
```

Zuweisen von SYSDTA. Der nächste Parameter wird von dieser Datei gelesen.

`Name` Name der Datei von RDATA zu lesen ist. (SYSDTA-Datei).

Dieser Parameter kann öfters angegeben werden. Statt "Name" kann auch *PRIMARY oder *SYSCMD angegeben werden.

Es wird intern das Kommando

`/ASSIGN-SYSDTA TO-FILE=Name`

abgesetzt.

4.2.4 Bedienung während des Betriebes

Die Bedienung erfolgt über das BS2000-Kommando '/SEND-MESSAGE'. Es gibt auch die Möglichkeit, von einem angeschlossenen CIS-Programm mit dem CIS-Kommando '$D' die SEND-MESSAGE Schnittstelle zu simulieren.

Übersicht über die Kommandos

Kommando	Bedeutung
/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))], MESSAGE=ENA_x $D_DENA_x[,DB.xxxxxx]	Behandlung der ENA-Comm (enable communication)
/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))], MESSAGE=END_b $D_DEND_b[,DB.xxxxxx]	Setzen der Beendigungs-bedingung von CISDBH
/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))], MESSAGE=DINFC[_p] $D_DINFC[_p][,DB.xxxxxx]	Information über Partner
/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))], MESSAGE=INFT[_p] $D_DINFT[_p][,DB.xxxxxx]	Informationen über Trans-aktionen
/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))], MESSAGE=DNTASKS_n $D_DNTASKS_n	Setzen der Anzahl Tasks

```
$D_DENA_x[,DB.xxxxxx]
```

Steuerung von ENA-COMM.

$D Operation für "Diagnose".

DENA Operationsergänzung für Steuerung von ENA-COMM (enable communication).

x
- L (Lock) : ENA-COMM sperren.
- U (Unlock) : ENA-COMM wieder freigeben.
- D (Display) : den aktuellen Zustand ausgeben.

Wird ENA-COMM gesperrt, so kann sich kein neuer Partner an CISDBH anschliessen.

xxxxxx Die Angabe eines Paßwortes DB.xxxxxx wirkt sich erst auf das nächste Kommando aus.

Quittung am Terminal: CISDBH - O.K. wenn Kommando ausgeführt wurde.

CISDBH - STATE: xxxx

bei Anforderung der Auflistung mögliche Werte:

OK	: alles O.K.
OKLE	: O.K. aber ENA gesperrt
ENDE	: Sitzung-beendet
LADE	: CISDBH wird geladen
KILL	: Kill-Kommando gegeben

```
$D_DEND_b[,DB.xxxxxx]
```

Setzen der Beendigungsbedingung von CISDBH.

$D Operation für "Diagnose".

DEND Operationsergänzung für Beendigungsbedingung.

b Beendigungsbedingung:
- 1 = CISDBH wird nach Abmeldung des letzten Partners beendet.
- 2 = CISDBH kann während der Betriebszeit nicht beendet werden. Außerhalb der Start- und Endezeit wird CISDBH auf die Beendigungsbedingung 1 gesetzt und nach Abmeldung des letzten Partners beendet.
- 3 = CISDBH kann nicht beendet werden. Zur Beendigung muß mit einem nachfolgenden Kommando die Beendigungsbedingung auf 1 oder 2 gesetzt werden.

xxxxxx Die Angabe eines Paßwortes DB.xxxxxx wirkt sich erst auf das nächste Kommando aus.

Quittungen am Terminal:
- CISDBH - O.K. wenn neue Beendigungsbedingung gesetzt wurde.
- CISDBH - ERROR wenn falsche Beendigungsbedingung angegeben wurde.

```
$D_DINFC[_p][,DB.xxxxxx]
```

Information über Partner.

$D Operation für "Diagnose".

DINFC Operationsergänzung für Information über Partner.

p Position in der internen Transaktionstabelle, ab der protokolliert wird. Wird p nicht angegeben, wird immer ab Position 1 ausgegeben.

xxxxxx Die Angabe eines Paßwortes DB.xxxxxx wirkt sich erst auf das nächste Kommando aus.

Beispiel einer Auflistung:

```
---- INFO-COMM. --DBHID=A -------2 0700 1800----------010/004-1-164431
-POS -TSN USER-ID.--MP-ADR ----SIZE COMM MODE ABBLINAM HOSTNAME --TIME
     3472 VD5DB   000B0000                    CISDBH   HOST1
     3471
     3473
     3474
   3 3494 VD5DB   000C3000 00000009 P    N 11          *        164325
```

Erläuterung der Überschriftszeile:

DBHID=A	Kennung des Data-Base-Handlers.
2	Beendigungsbedingung.
0700	Startzeit
1800	Endezeit
010/004-1	Es können maximal 10 Tasks vorhanden sein, zur Zeit gibt es jedoch nur vier. Die Angabe -1 zeigt, daß von den 4 Tasks einer ein Verwaltungstask ist.
164431	Uhrzeit der Protokollierung (Stunde, Minuten, Sekunden).

Erläuterung der Datenzeilen:

POS	Position in der internen Tabelle der Partner. Die ersten Zeilen betreffen immer den Data-Base-Handler selbst und haben keine Positionsnummern. Nach der letzten Datenzeile erscheint: END (xxx). xxx = max. Anzahl der angeschlossenen Partner. (siehe auch Parameter DNCOMM).
TSN	Task-Sequence-Number des Partners (bzw. Data-Base Handlers in den ersten Zeile). Beim Multitask-Betrieb werden in den ersten Zeilen die TSNs der DBHs aufgelistet. In der 1. Zeile steht immer die TSN des Tasks, der die Information ausgegeben hat. Wenn es einen Verwaltungstask gibt, so steht seine TSN immer in der 2. Zeile.
USER-ID	Benutzerkennung des Partners (bzw. Data-Base-Handlers in der 1. Zeile).
MP-ADR	Adresse des Bereichs für diesen Partner im Common-Memory-Pool. Verkehrt der Partner über DCAM (siehe Spalte COMM), so steht hier die Adresse des Bereichs, in dem die Nachrichten gespeichert werden.
SIZE	Bei P1-Eventing (P in Spalte COMM): Größe des Memory-Pool-Bereichs in Vielfachem von 4 K. Bei DCAM (D in Spalte COMM): 4 Stellen vorgegebene Elementlänge (Konstante in CISCON=X'07D0'=2000) 4 Stellen maximale Länge für diese Verbindung
COMM	Art der Kommunikation: P = über P1-Eventing und Common-Memory-Pool D = über DCAM
MODE	Betriebsart des Partners: N = Teilnehmer (TIAM) S = UTM-Synchronisiert Hinter der Betriebsart steht die CIS-Version des Partners.

APPLINAM Name der Anwendung: 1. Zeile: (Data Base Handler): leer, wenn kein DCAM-Anschluß: (DCAM=N).

CISDBHx, wenn DCAM-Anschluß. x = Data Base Handler Kennung.

Weitere Zeilen (Partner): Der Name der Anwendung. Er hängt von der Betriebsart und der Art der Kommunikation ab:

COMM	MODE	APPLINAM
P	N	Spaces
P	S	Name der UTM-Anwendung vgl.: MAX APPLINAME=... für KDCDEF
D	N	CISTttt tttt = TSN des Partners
D	S	Name der UTM-Anwendung

HOSTNAME Name des Rechners
1. Zeile (Data-Base-Handler): Leer, wenn kein DCAM-Anschluß
Rechnername, wenn DCAM-Anschluß.

weitere Zeilen (Partner): Der Name des Rechners. Er hängt von der Betriebsart und der Art der Kommunikation ab:

COMM	MODE	HOSTNAME
P	N	*
P	S	Name des Rechners
D	alle	

TIME Uhrzeit des Verbindungsaufbaus (Stunden, Minuten, Sekunden). In der 1. Zeile steht die Startzeit des DBHs.

```
$D_DINFT[_p][,DB.xxxxxx]
```

Information über Transaktionen.

$D	Operation für "Diagnose".
DINFT	Operationsergänzung für Information über Transaktionen.
p	Position in der internen Transaktionstabelle, ab der protokolliert wird. Wird p nicht angegeben, so wird immer ab Position 1 ausgegeben.
xxxxxx	Die Angabe eines Paßwortes DB.xxxxxx wirkt sich erst auf das nächste Kommando aus.

Beispiel einer Auflistung:

```
---- INFO-TRANS-CISDBH ------------------------------------------
-POS HOSTNAME APPLINAM ----USER -------------TID -TSN MODE  -PAGE
   1 *                  4687    4687             4687 N     A0042
   2 GZ432050 CISUTM    ADMIN   0100000100010019 4687 S     A0047
END  (100)
```

Erläuterung der Datenzeilen:

POS — Position in der internen Tabelle der Transaktionen.

Nach der letzten Datenzeile erscheint: END (xxx)

xxx = max. Anzahl der aktuellen Transaktionen. (siehe auch Parameter DNTRANS).

HOSTNAME — Name des Rechners. Der Name des Rechners hängt von der Betriebsart und der Art der Kommunikation ab:

COMM	MODE	HOSTNAME
P	N	*
P	S	Name des Rechners
D	alle	

APPLINAM Name der Anwendung. Der Name der Anwendung hängt von der Betriebsart und der Art der Kommunikation ab:

COMM	MODE	APPLINAM
P	N	Spaces
P	S	Name der UTM-Anwendung vgl.: MAX APPLINAME=... für KDCDEF
D	N	CIStttt tttt = TSN des Partners
D	S	Name der UTM-Anwendung

USER

MODE	USER
S	KCBENID / KCLOGTER
sonst	tsn

TID Transaktionskennung

MODE	TID
S	interne UTM-Transaktionskennung
sonst	tsn des Partners

TSN Task-Sequence-Number des Partners.

MODE Betriebsart des Partners: N = Teilnehmer (TIAM)
S = UTM-synchronisiert

PAGE Interne Nummer. Zeigt wo die Daten dieser Transaktion gespeichert werden:

Aaaaa: Nr. der Seite im Memory-Pool
Ppppp: Nr. der PAM-Page in der Datei CISUTM.RIOFILE.CISDBHi

```
$D_DNTASKS_n[,DB.xxxxxx]
```

Setzen der Anzahl Tasks.

$D	Operation für "Diagnose".
DNTASKS	Operationsergänzung für Setzen Anzahl Tasks.
n	Neue Anzahl der Tasks. Ist n größer als der bisherige Wert, so werden weitere Tasks gestartet. Hierzu werden im CISDBH ENTER-JOB Kommandos abgesetzt. Der CISDBH-Parameter DENT-JOB muß vorhanden sein. Ist n kleiner als der bisherige Wert, so werden Tasks beendet, bis nur noch die neue Anzahl Tasks aktiv sind. Die Tasks werden nur nach dem Ende eines Zyklus beendet. n kann nicht größer als der Maximalwert (DMAXT=) werden, und auch nicht kleiner als 1. Wird einer dieser Werte angegeben, so wird ohne Fehlermeldung max. bzw. 1 angenommen. Wenn der 1. Task ein Verwaltungstask ist, kann n nicht kleiner als 2 werden.
xxxxxx	Die Angabe eines Paßwortes DB.xxxxxx wirkt sich erst auf das nächste Kommando aus.

Quittungen am Terminal:	CISDBH - O.K.	wenn Kommando ausgeführt wurde.
	CISDBH - ERROR	nach Fehler bei /ENTER-JOB.

4.2.5 Beendigung von CISDBH

Die Beendigung des CISDBH wird über die Beendigungsbedingung und die eingestellte Betriebszeit (vgl. Seite 159 und 163) gesteuert.

Beendigungsbedingung:

1 = CISDBH wird nach Abmeldung des letzten Partners beendet.

2 = CISDBH kann während der Betriebszeit nicht beendet werden. Außerhalb der Start- und Endezeit wird CISDBH auf die Beendigungsbedingung 1 gesetzt und nach Abmeldung des letzten Partners beendet.

3 = CISDBH kann nicht beendet werden. Zur Beendigung muß mit dem Kommando $D DEND b (vgl. Seite 167) die Beendigungsbedingung auf 1 oder 2 gesetzt werden.

Folgende Prozeßschalter können nach Beendigung abgefragt werden:

25=ON Eine CIS-Fehlermeldung trat während des CISDBH-Laufs auf.

26=ON Abbruch von CISDBH durch einen Fehler in CISDBH mit x'00' < IW < x'80'.

27=ON Abbruch von CISDBH durch einen Fehler in CIS. CISDBH kann neu geladen werden (z.B. SKIP-COMMANDS Kommando in der Start-Prozedur).

4.2.6 Dateinamen

CISDBH benützt folgende Dateinamen:

Datei	LINK-Name	Standard-Dateiname
DABEL	DB	DABEL
Hauptdateien	1)	
Verweisdateien	1)	
ISISOUT-Datei	ISISOUT	
After-Image-Datei 2)	AIM	CIS.AIM
Before-Image-Datei 2)	BIM	CIS.BIM
Protokoll-Datei 2)	PRO	CIS.PRO
Rollin / Rollout-Datei 3)		CISUTM.RIOFILE.CISDBHi

1) Wird intern wie beim eingebundenen CIS vergeben.
2) Nur wenn der eingebundene CISKOOR auf diese Dateien zugreift.
3) von CISDBH eingerichtet und verwaltet.

Größe der Rollin-/Rollout-Datei:

Die Größe (in Anzahl PAM-Pages) errechnet sich mit folgender Formel:

(DNTRANS Wert - DNRIO Wert) * 10

Somit ergibt sich mit den Defaultwerten eine Größe von: (200 - 50) * 10 = 1.500 Pages

4.3 Kommunikation zu CISDBH

4.3.1 Allgemeines

Die Kommunikation zwischen dem Modul CISCON (eingebunden in dem Programm, das CISDBH aufruft) und CISDBH kann auf zwei Arten erfolgen:

1. DCAM=N - Kommunikation über P1-Eventing und Common-Memory-Pool.

   Die Technik ist ähnlich der Kommunikation zu CISKOOR. Diese Art der Kommunikation ist nur möglich, wenn beide Programme im gleichen Rechner sind.

   Es werden zwei Möglichkeiten des Betriebssystems genutzt:

   - P1-Eventing: Überträgt Signale, meldet also, daß eine Nachricht vorhanden ist.
   - Common-Memory-Pool: Übergibt die Daten.

2. DCAM=J/Y - Kommunikation über DCAM.

   Diese Kommunikation ist möglich, wenn beide Programme auf dem gleichen Rechner oder auf zwei verschiedenen Rechnern ablaufen.

Die ganze Kommunikation läuft automatisch, der Benutzer muß lediglich die Größe des Common Memory-Pools festlegen.

4.3.2 Benützung des Memory-Pools

Der Memory-Pool wird von CISDBH eingerichtet. CISDBH legt seine Größe fest, sie kann während des Laufes nicht mehr verändert werden.

Findet CISDBH beim Hochfahren einen schon besteheneden Memory-Pool vor, nimmt er keine Nachrichten an. Mit dem BS2000 Kommando '/SEND-MESSAGE' kann festgestellt werden, wer an dem Memory-Pool angeschlossen ist. Dieser Fall tritt auf, wenn der letzte Task einer DBH-id abgebrochen (z.B. CANCEL) wurde und noch mindestens ein Partner-Task, der auch an diesen Memory-Pool angeschlossen war, vorhanden ist. Das BS2000 gibt den Memory-Pool nicht frei, solange noch mindestens ein Task angeschlossen ist. Damit CISDBH in diesem Fall wieder geladen werden kann, müssen alle Programme, die mit CISDBH arbeiten, beendet werden.

Sofort nach dem Einrichten des Memory-Pools belegt CISDBH einen Teil am Anfang des Pools. Die Größe dieses Teils errechnet sich nach der Formel:

Größe CISDBH-Teil = 4096 + m*8 + p*(48+(m*4)) (in Bytes) aufgerundet auf ein Vielfaches von 4 K.

m = Maximale Anzahl der Tasks für CISDBH (vgl. DMAXT-Parameter - Seite 160).

p = Anzahl der Partner für CISDBH (vgl. DNCOMM-Parameter - Seite 161).

Der standardmäßig ausgelieferte CISDBH benötigt 16 K.

Jeder Task, der mit CISDBH Verbindung aufnimmt, belegt 36 K im Memory-Pool. Hierüber tauscht er Informationen mit CISDBH aus.

4.3.3 Berechnen der Memory-Pool Größe

Die Größe des Memory-Pools errechnet sich nach der Formel:

Größe Memory-Pool = d + n * 36 (in KB)

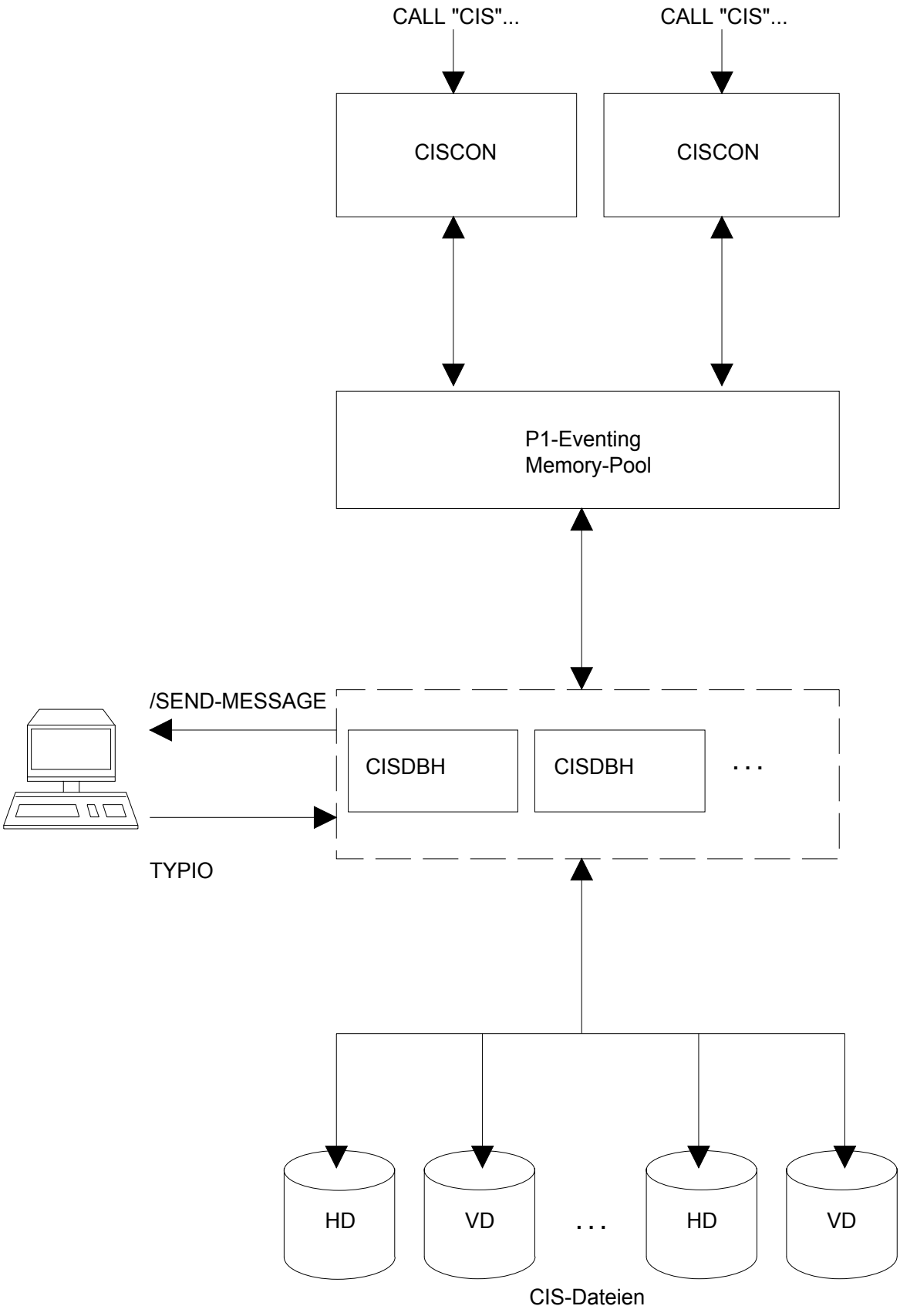
d = Von CISDBH benötigter Platz (vgl. Seite - 175)

n = Maximale Anzahl Partner von CISDBH

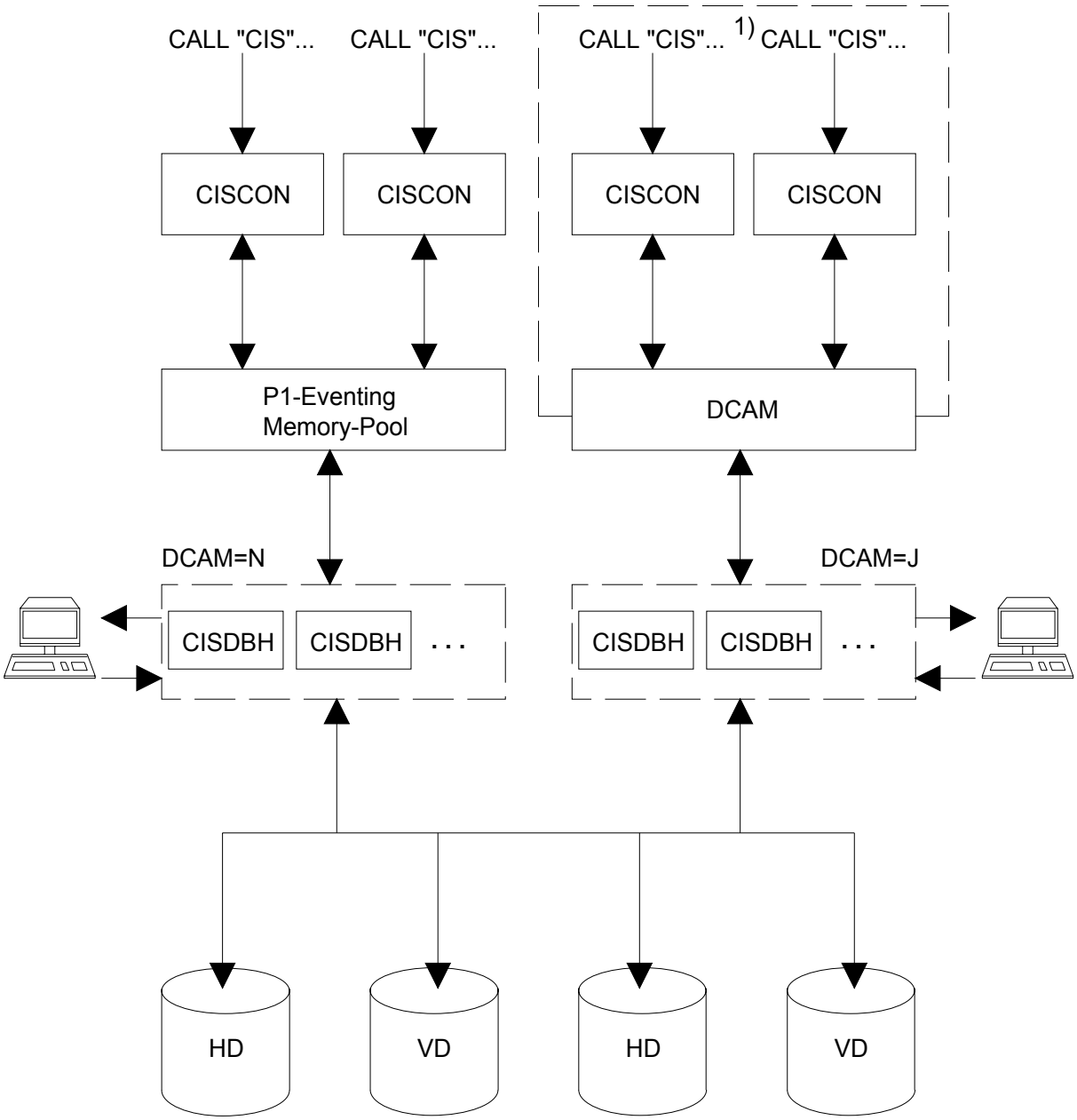
Die Größe des Memory-Pools wird aufgerundet auf ein Vielfaches von 64 KB.

4.4 Datenflußpläne von CISDBH

Datenfluß ohne Datensicherung - Über P1-Eventing

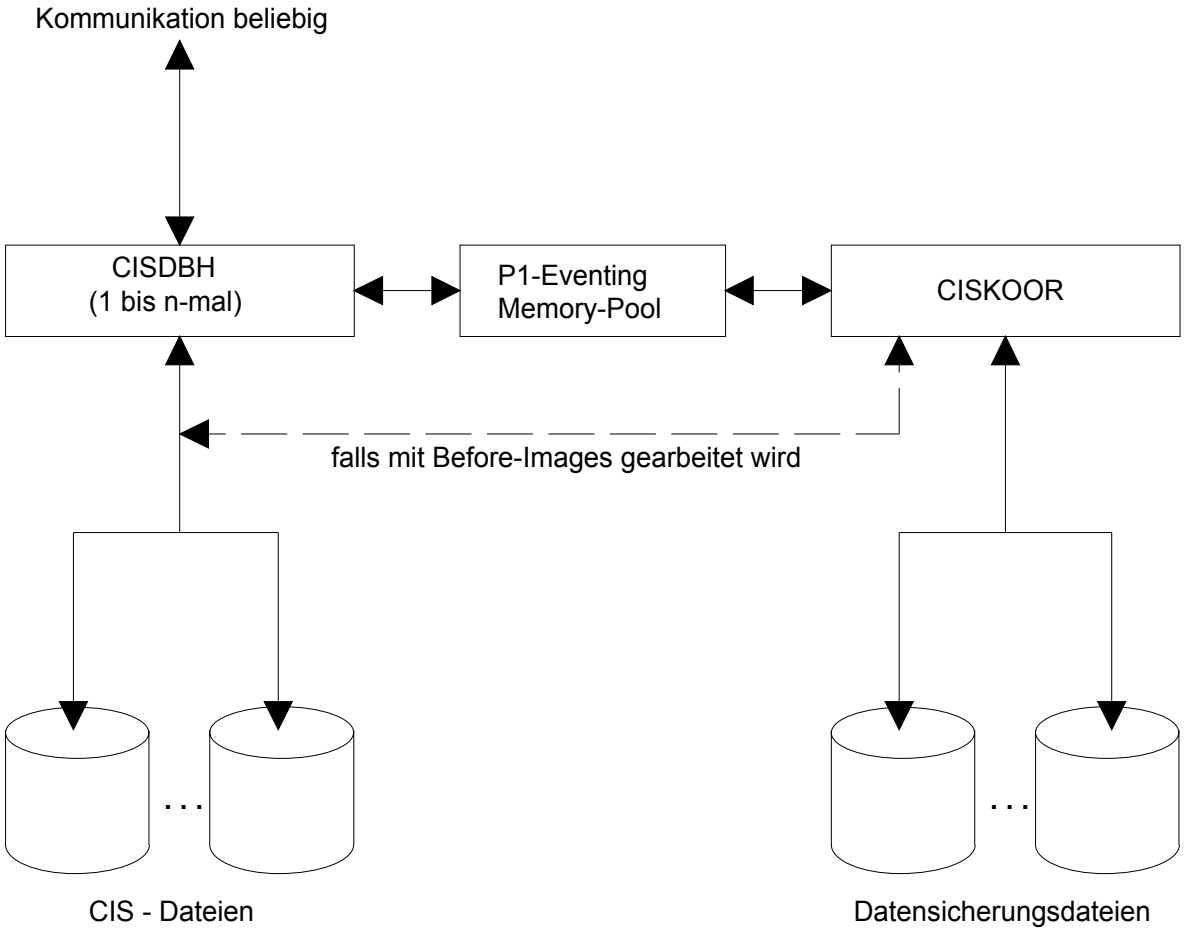


Datenfluß ohne Datensicherung - Über P1-Eventing oder DCAM

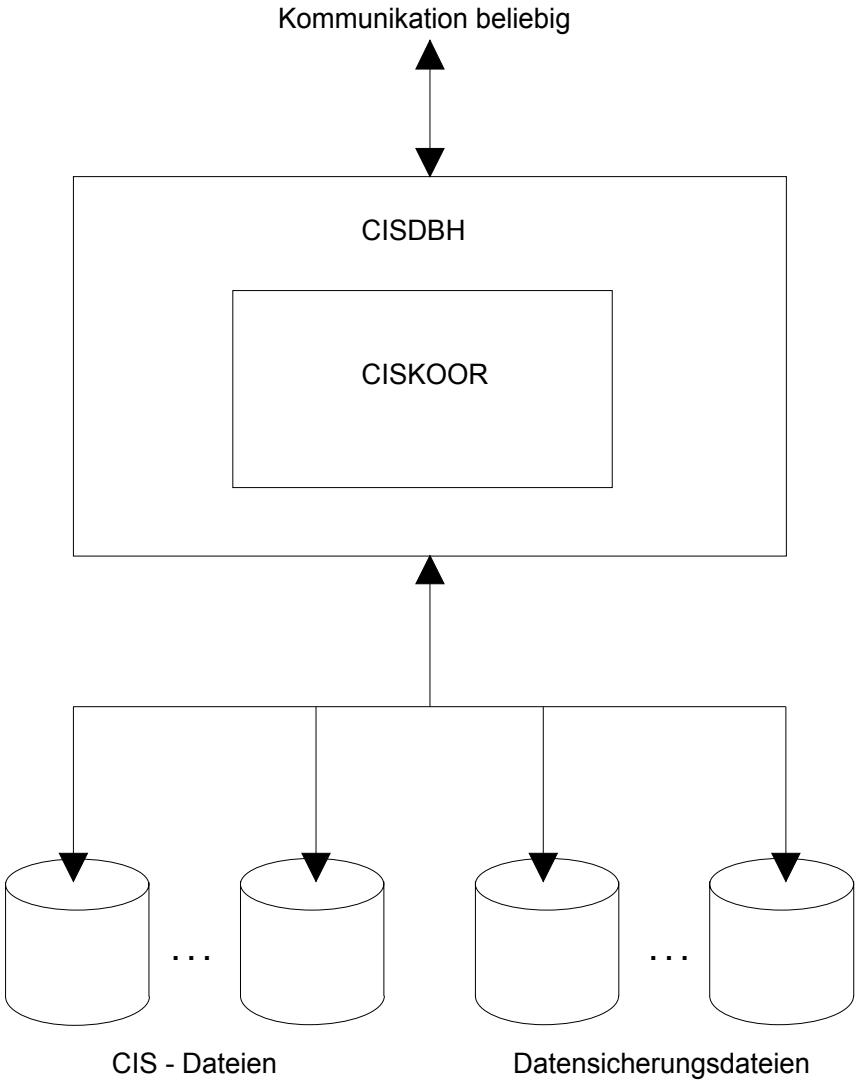


1) dies kann in einem anderen Rechner sein

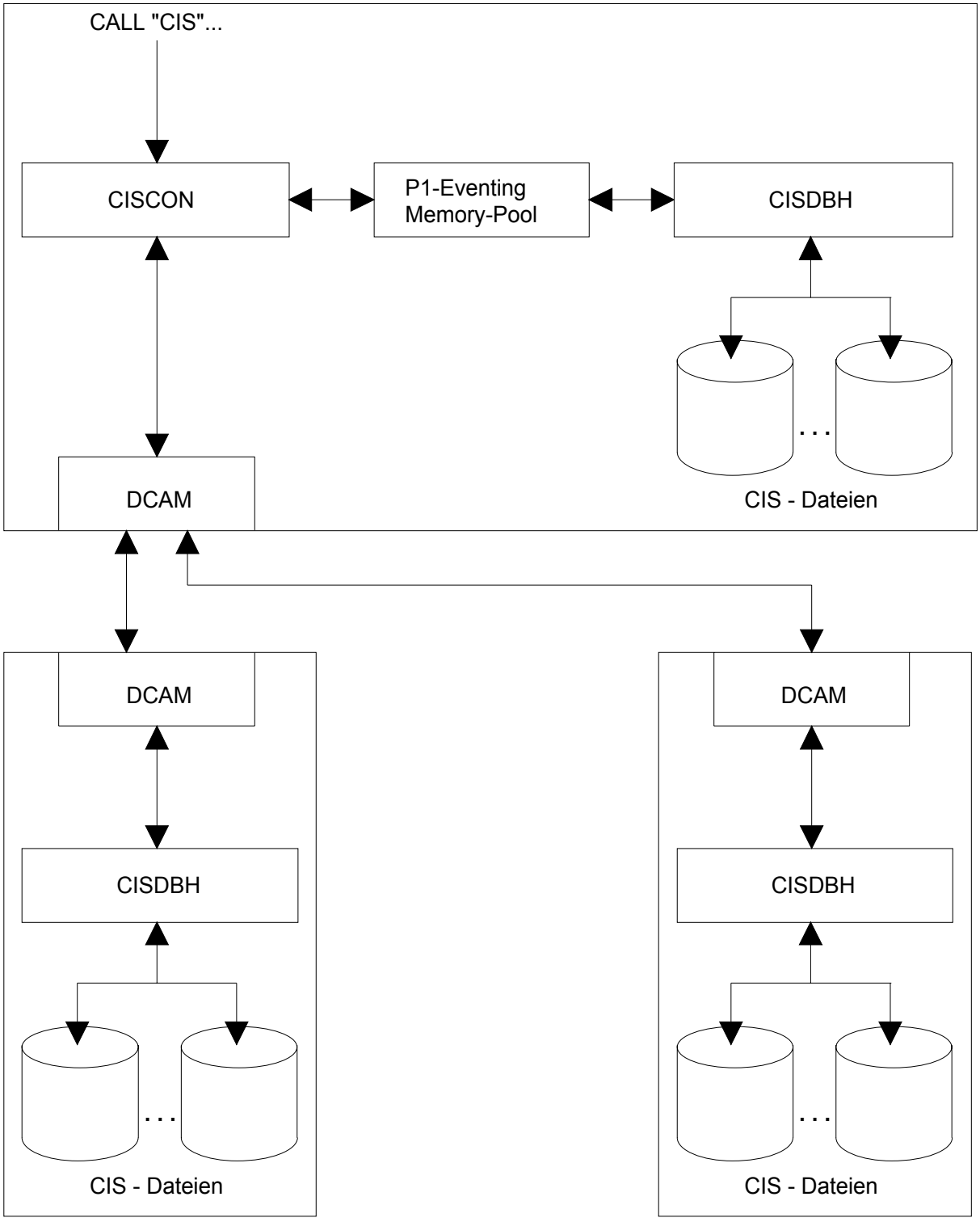
Datenfluß mit Datensicherung (unabhängiger CISKOOR)



Datenfluß mit Datensicherung (eingebundener CISKOOR)



Rechnerverbund



4.5 Aufruf von CISDBH (über CISCON)

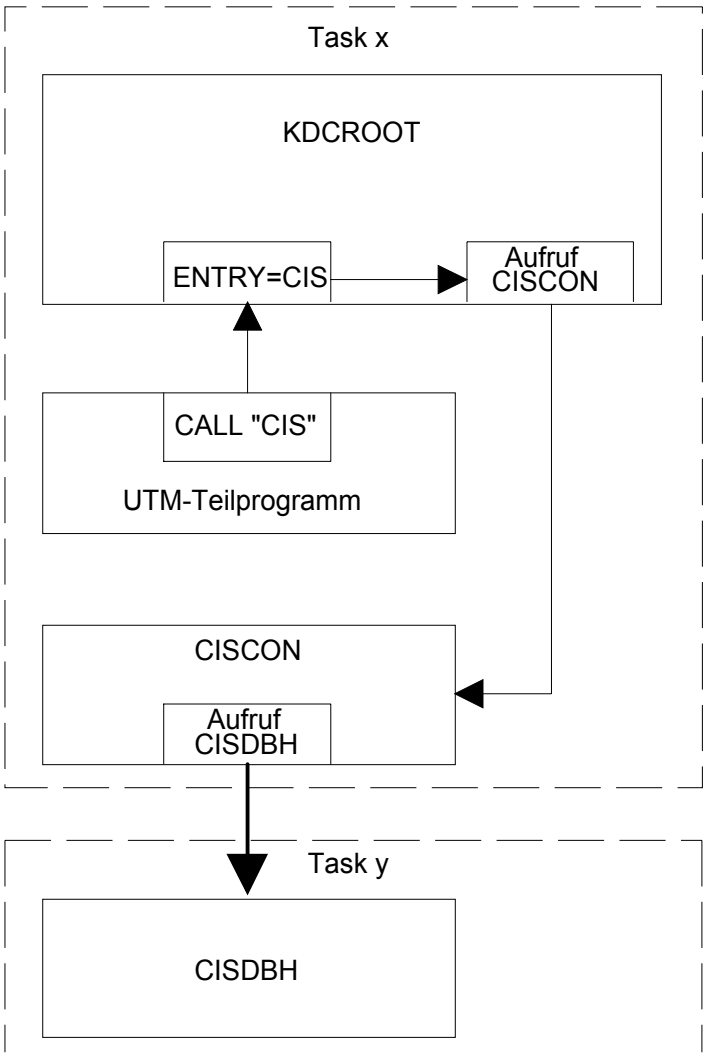
Die Benutzerprogramme rufen CIS über den ENTRY "CIS" in CISCON auf. Über das Modul CISCON werden die Parameter an CISDBH weitergegeben, die Antwort empfangen und an das aufrufende Programm zurückgegeben. Die Verbindung zwischen dem Modul CISCON und dem Programm CISDBH ist über P1-Eventing oder DCAM realisiert. Das Modul CISCON braucht Parameter, z.B. zum Steuern der Verbindungen zu den CISDBHs.

4.5.1 Betriebsarten

Es gibt zwei mögliche Betriebsarten:
- synchronisierter UTM-CIS-Betrieb
- TIAM-Betrieb

Synchronisierter UTM-CIS-Betrieb

Die Übersicht zeigt die Zusammenarbeit der einzelnen Module:



Zusätzlich zu den CIS-Aufrufen, die das vom Benutzer geschriebene UTM-Teilprogramm absetzt, gibt es noch CIS-Aufrufe von UTM.

Die Parameter für CISCON werden in der Startroutine von KDCROOT gelesen und an CISCON übergeben. Um sie von den UTM-Parametern unterscheiden zu können, muß in den ersten fünf Stellen der Name ".CIS_" stehen. Der CISCON-Parameter END entfällt.

Die CISCON-Parameter werden gemeinsam mit den UTM-Parametern vor END eingegeben.

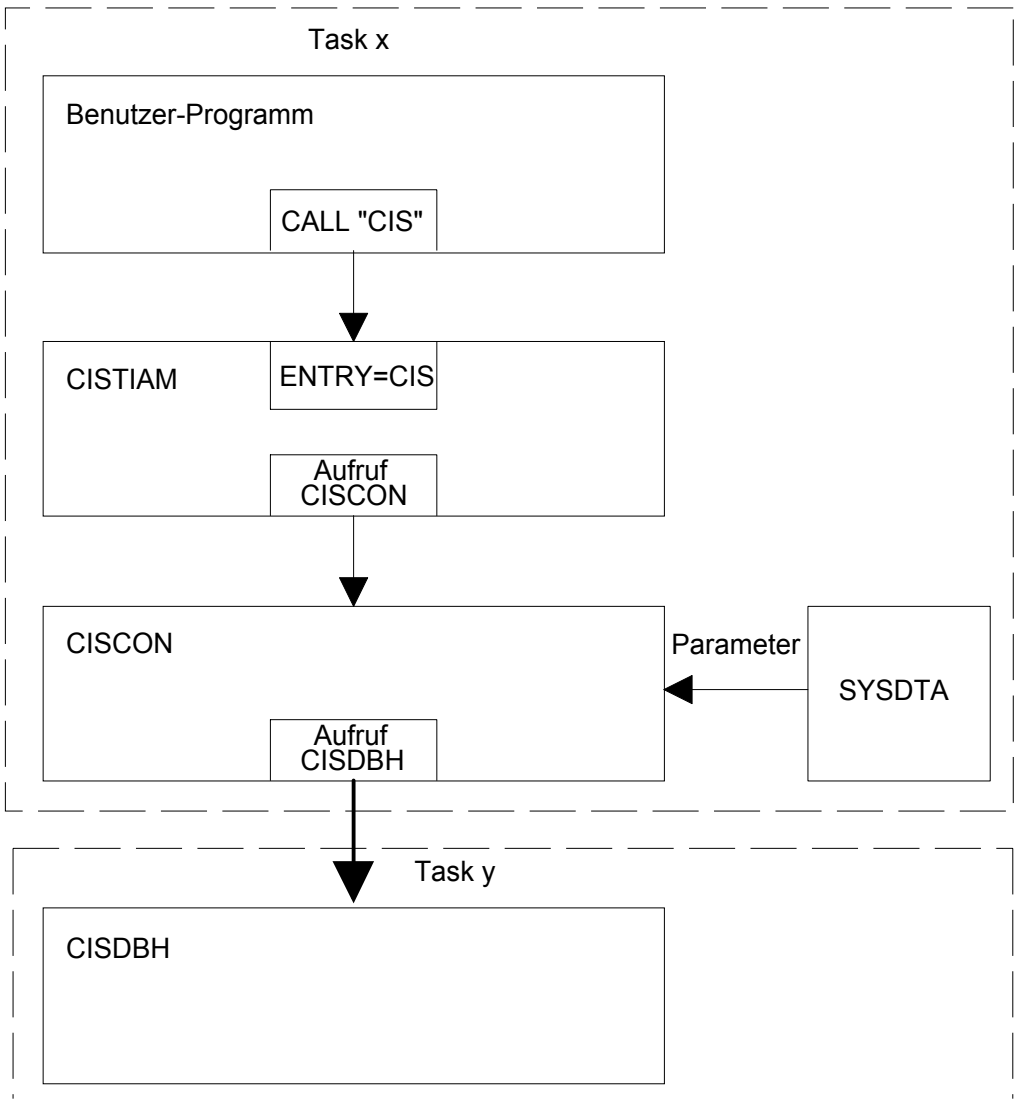
z.B. .

```
.
.
.
.UTM START FILEBASE=...
.UTM START TASKS=...
.CIS DBH=*/A
                                    } CISCON-Parameter
.CIS ENT-JOB=E.CISDBH.A
.UTM END
```

Für diese Betriebsart muß bei der Generierung mit KDCDEF der Parameter DATABASE ,TYPE=CIS angegeben werden. Das zum Generieren von KDCROOT benötigte Makro KDCDB wird mit CIS ausgeliefert. Ab UTM V3.2 wird das Makro KDCDBC benötigt, wenn 2 Datenbank-Systeme mit UTM zusammenarbeiten.

TIAM-Betrieb

Die Übersicht zeigt das Zusammenwirken der einzelnen Module:



Die CISCON-Parameter werden beim ersten CIS-Aufruf über SYSDTA gelesen. Der letzte Parameter muß END sein.

In den CISCON-Parametern kann der Name .CIS angegeben werden.

Wenn CISCON-Parameter aus verschiedenen Quellen eingelesen werden sollen, wird SYSDTA mit dem SYSDTA-Parameter zugewiesen. Wenn das Benutzerprogramm aus SYSDTA lesen soll, kann nach dem Lesen der CISCON-Parameter mit dem END-Parameter die SYSDTA-Zuweisung geändert bzw. rückgesetzt werden.

Das Benutzerprogramm kann vor dem ersten eigentlichen CIS-Aufruf mit dem CIS-Kommando, `CISCON <parameter>`, Parameter an CISCON übergeben. Alle Parameter sind zulässig. Als letzter Parameter muß END eingegeben werden.

z.B.:
```
CISCON DBH=*/A
CISCON END
```

Dann wird CIS mit dem eigentlichen Kommando aufgerufen. Die Verbindung zu CISDBH wird aufgebaut.

Beispiel:

Das Programm CISIND liest CIS-Kommandos vom Terminal, ruft CIS auf und gibt die Antwort wieder am Terminal aus. CISIND setzt einen Aufruf an CIS ab, damit die CISCON-Parameter eingelesen werden. Mit folgender Prozedur kann dieses Programm gestartet werden:

```
/BEGIN-PROCEDURE

/SET-FILE-LINK FILE-NAME=Name,LINK-NAME=DB
                                        Zuweisung der DABEL (mit den Fehler-
                                        texten).

/ASSIGN-SYSDTA TO-FILE=*SYSCMD          SYSDTA auf SYSCMD zuweisen.

/START-PROGRAM FROM-FILE=CISIND         Programm laden.

ENT-JOB=E.CISDBH.A                      Parameter für automatischen ENTER.

DBH=*/A                                 Auswahl des CISDBH.

END *PRIMARY                            Ende der CISCON-Parameter und
                                        zurückschalten von SYSDTA auf das
                                        Terminal.

/END-PROCEDURE
```

4.5.2 Parameter für CISCON

Übersicht:

Parameter	Bedeutung
`*...`	Kommentar
`CISVARI+D={C'c...c'/X'xx..xx'}`	Korrektur von CISVARI (Parameter für Programmablauf)
`DBH=r/i`	Angabe eines CISDBH
`DIST=pppppp,r/i`	Angabe einer Verteilregel
`END[_Parameter]`	Ende der Parameter
`ENT-JOB=Parameter`	Parameter zum Starten CISDBH-Task
`PARTNER=r/i`	Definition eines DBH mit DPASS-Parameter
`SYSDTA=Name`	Umschalten von SYSDTA (Parametereingabe)
`WAIT=nnnn`	Wartezeit (in Sek.) auf Antwort von CISDBH

Jeder Parameter kann einen Kommentar enthalten. Der Kommentar wird durch mindestens ein Space von den Parameterwerten getrennt.

```
*...
```

Kommentar: Es können beliebig viele Kommentare vorhanden sein.

```
CISVARI+d={C'c...c'/X'xx..xx'}
```

Korrektur von CISVARI (variable Parameter für die Ablaufumgebung von CIS)

`d` Distanz CISVARI (sedezimal anzugeben, maximal 4-stellig)

`c...c` Text für CISVARI

`xx...xx` Sedezimaler Text für CISVARI (immer gerade Zahl von Zeichen)

Es wird geprüft, ob der Text nicht über CISVARI hinausgeht. Es können mehrere CISVARI-Parameter angegeben werden.

```
DBH=r/i
```

Dieser Parameter wird angegeben, wenn nur mit einem Data-Base-Handler Verbindung aufgenommen wird und wenn dieser Data-Base-Handler keine DPASS-Parameter hat. Ein Data-Base-Handler kann auch ein Multi-Task DBH sein. Er muß über eine einzige DBH-Kennung angesprochen werden.

Dieser Parameter darf nur einmal vorhanden sein. Bei Angabe des Parameters darf auch kein DIST- und kein PARTNER-Parameter angegeben werden. Alle CIS-Kommandos werden an diesen Data-Base-Handler weitergegeben.

`r` Rechner, in dem der Data-Base-Handler läuft. Wird * angegeben, ist der Data-Base-Handler im eigenen Rechner und die Verbindung geht über P1-Eventing. Wird ein 1- bis 8-stelliger Name angegeben, geht die Verbindung über DCAM. Im Data-Base-Handler muß dann der Parameter DCAM=Y/J angegeben sein.

`i` Identifikation des Data-Base-Handler (1 oder 2 Byte). Entspricht dem Parameter DBHID=i von CISDBH.

```
DIST=pppppp,r/i
```

Dieser Parameter ordnet ein CIS-Paßwort einem bestimmten Data-Base-Handler zu. Bei Verwendung des DIST Parameters darf der DPASS Parameter nicht angegeben werden.

Der Parameter wird für jedes CIS-Paßwort angegeben, das bearbeitet werden soll. Anhand dieser Parameter wird eine Tabelle aufgebaut, die angibt, welcher Data-Base-Handler welches Paßwort bearbeitet.

In diesem Betriebsmodus muß das erste CIS-Kommando jeder Transaktion ein CIS-Paßwort enthalten, da nach dem Paßwort verteilt wird.

Dieser Parameter kann zusammen mit dem PARTNER-Parameter (vgl. Seite 190), jedoch nicht mit dem DBH-Parameter (vgl. Seite 187) angegeben werden. Für einen bestimmten CISDBH dürfen jedoch nur DIST-Parameter oder ein PARTNER-Parameter angegeben werden.

`pppppp` CIS-Paßwort (immer 6-stellig). Die CIS-Paßworte müssen im Netz der verschiedenen Data-Base-Handler eindeutig sein.

`r` Rechner, in dem der Data-Base-Handler läuft.

`i` Identifikation des Data-Base-Handlers (1 oder 2 Byte). Entspricht dem Parameter DBHID=i von CISDBH.

```
END[_Parameter]
```

Ende der Parameter

Wird noch ein Operand hinter END angegeben, so wird dieser in ein '/ASSIGN-SYSDTA TO-FILE=' Kommando eingefügt. Bei Angabe von *PRIMARY. wird SYSDTA wieder auf die ursprüngliche Zuweisung zurückgesetzt. Dies kann wichtig sein, wenn der SYSDTA-Parameter benützt wurde.

Im synchronisierten UTM-Betrieb wird dieser Parameter nicht angegeben

```
ENT-JOB=Parameter
```

Parameter zum Starten CISDBH-Task (automatischer ENT-JOB).

`Parameter` Parameter für ENT-JOB Kommando (maximal 200 Bytes).
Vgl. BS2000-Kommando /ENTER-JOB: Der Parameter wird ab dem Operanden "Dateiname" angegeben. (Es wird also /ENTER-JOB FROM-FILE = davor gesetzt.)

Im Dateinamen muß der String "CISDBH.i" (siehe auch Seite 153) vorhanden sein; i ist die Identifikation des Data-Base-Handlers (siehe DBHID=i).

Beim Multitask-DBH wird der 1. Task von CISCON gestartet. Die weiteren Tasks werden dann vom 1. DBH gestartet (vgl. CISDBH-Parameter DNTASKS und DENT-JOB, Seite 172 und 159).

Dieser Parameter darf nur einmal angegeben werden. Ist er nicht vorhanden, wird nicht mit automatischem ENT-JOB gearbeitet.

Sollen mehrere Data-Base-Handler automatisch gestartet werden, müssen die Namen der ENTER-Dateien folgender Norm entsprechen:

Beispiel: Die Data-Base-Handler mit den Kennungen A, B und I sollen automatisch geladen werden:

Es gibt folgende ENTER-Dateien:
```
E.CISDBH.A
E.CISDBH.B
E.CISDBH.I
```

Die CISCON-Parameter sind:
```
ENT-JOB=E.CISDBH.A
PARTNER=*/A
PARTNER=*/B
PARTNER=*/I
     .
     .
     .
```

Der Buchstabe A im Parameter ENT-JOB=E.CISDBH.A ... ist nur ein Platzhalter. Er wird jeweils durch die DBH-Kennung ersetzt. In diesem Fall also nacheinander durch A, B, I.

```
PARTNER=r/i
```

Dieser Parameter muß angegeben werden, wenn mit einem Data-Base-Handler, in dem der DPASS-Parameter (vgl. Seite 162) verwendet wird, Verbindung aufgenommen wird.

Beim Verbindungsaufbau meldet CISDBH, welche Paßworte verarbeitet werden. CISCON baut die Verteiltafel auf.

In diesem Betriebsmodus muß das erste CIS-Kommando jeder Transaktion ein CIS-Paßwort enthalten, da die Verteilung anhand des Paßwortes erfolgt.

Dieser Parameter kann zusammen mit dem DIST-Parameter (vgl. Seite 188), jedoch nicht mit dem DBH-Parameter (vgl. Seite 187) angegeben werden. Für einen bestimmten CISDBH dürfen jedoch nur (mehrere) DIST-Parameter oder ein PARTNER-Parameter angegeben werden.

`r` Rechner, in dem der Data-Base-Handler läuft (siehe DBH-Parameter).

`i` Identifikation des Data-Base-Handlers (1 oder 2 Byte). Entspricht dem Parameter DBHID=i von CISDBH.

```
SYSDTA=Name
```

Umschalten von SYSDTA (Parametereingabe)

`Name` Name der SYSDTA-Datei. Bei Angabe von *PRIMARY wird SYSDTA auf die ursprüngliche Zuweisung zurückgesetzt. Intern wird ein Kommando

```
/ASSIGN-SYSDTA TO-FILE=Name
```

abgesetzt.

```
WAIT=nnnn
```

`nnnn` 1- bis 4-stellige Zeitangabe

Mit dem WAIT-Parameter wird die Wartezeit (Angabe in Sekunden) auf eine Antwort von CISDBH gesetzt. Nach Ablauf der Zeit und Ausbleiben einer Antwort von CISDBH, wird der Fehlercode CO28 ausgegeben.

Wird der Parameter nicht angegeben, beträgt die Wartezeit 60 Sekunden.

4.5.3 Zusammenwirken der CISDBH- und der CISCON-Parameter (Verteilregel)

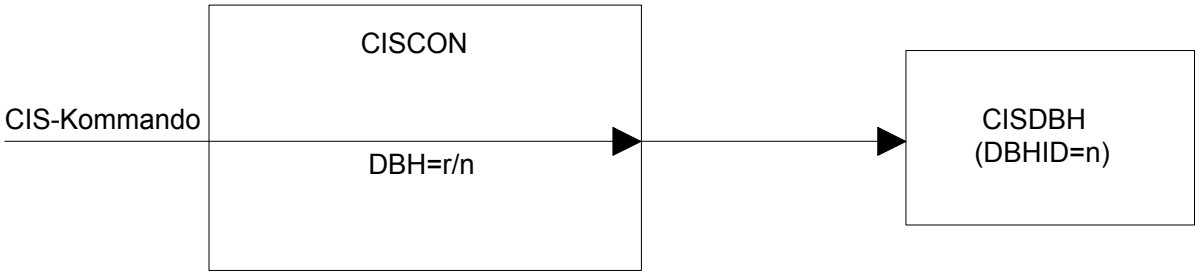
Fall	Verbindung mit	CISDBH Parameter	CISCON Parameter
1	nur 1 DBH		DBH=
2			DIST= . DIST= } immer gleicher Partner
3		DPASS=	PARTNER=
4	mehrere DBH		DIST= . DIST= } immer gleicher Partner
5		DPASS=	PARTNER=

Aus der Sicht von CISCON können Fall 4 und Fall 5 gleichzeitig vorhanden sein.

Verteilregel bilden

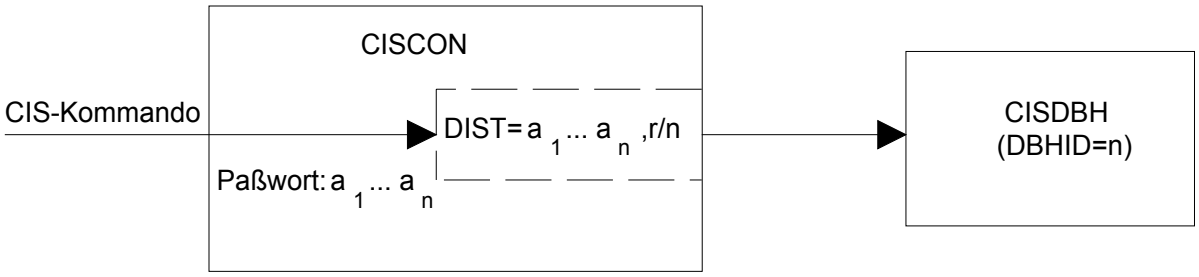
Folgende Diagramme veranschaulichen die obige Tabelle.

Fall 1:



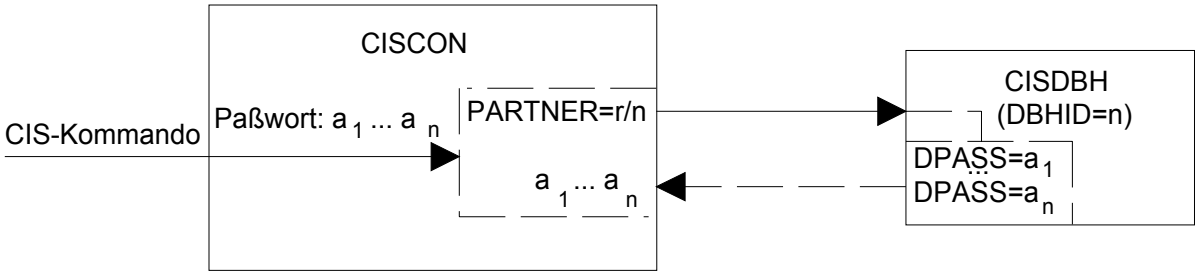
Die DBH-Anweisung teilt CISCON mit, daß alle Aufträge vom Data-Base-Handler 'n' auf Rechner 'r' bearbeitet werden.

Fall 2:



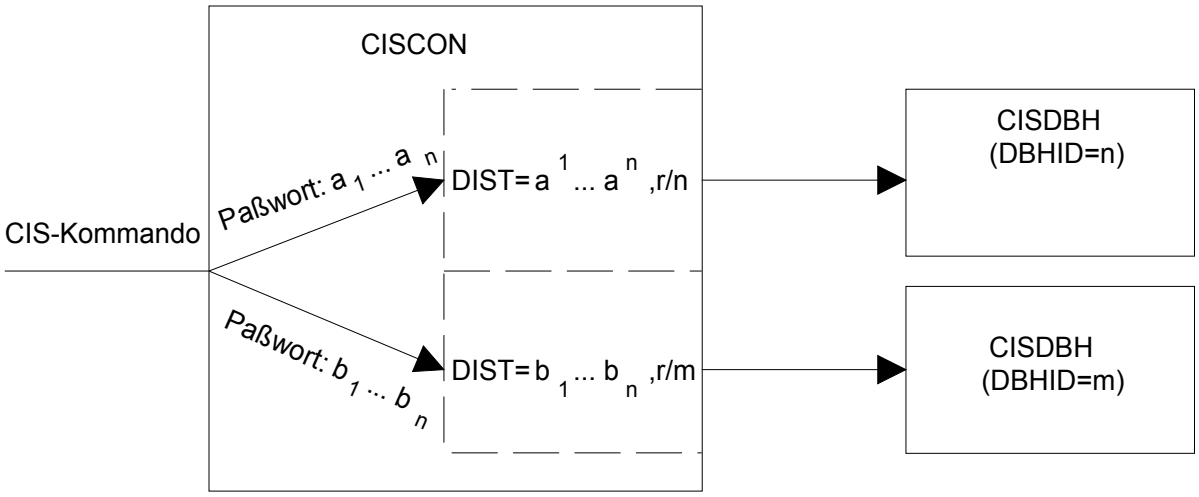
Die DIST-Anweisung teilt CISCON die Paßwörter mit, mit denen die Aufträge gekennzeichnet sein müssen, um vom Data-Base-Handler 'n' auf Rechner 'r' bearbeitet werden zu können. Aufträge, deren Paßwörter nicht übereinstimmen, werden von CISCON abgewiesen.

Fall 3:



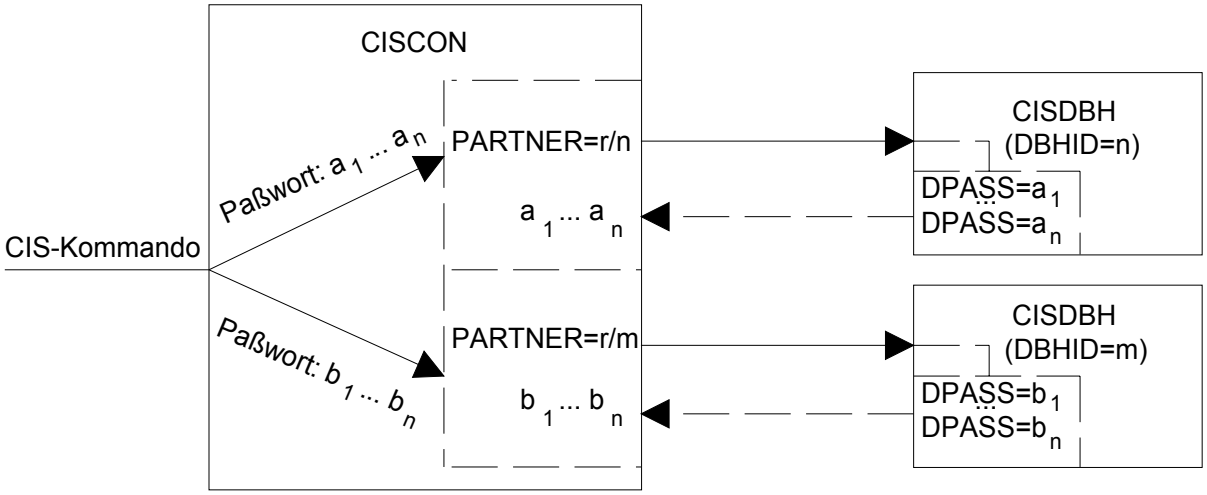
Die PARTNER-Anweisung teilt CISCON mit, daß mit dem Data-Base-Handler 'n' auf Rechner 'r' gearbeitet werden soll. Die DPASS-Anweisung teilt dem CISDBH die Paßwörter mit, mit denen die Aufträge gekennzeichnet sein müssen, um bearbeitet werden zu können. Bei der ersten Verbindungsaufnahme von CISCON mit CISDBH werden die Paßwörter an CISCON übergeben. Aufträge mit nicht übereinstimmenden Paßwörten werden von CISCON abgewiesen.

Fall 4:



Mehrere DIST-Anweisungen teilen CISCON mit, von welchem CISDBH auf einem bestimmten Rechner, Aufträge, versehen mit einem bestimmten Paßwort, bearbeitet werden sollen. Aufträge, deren Paßwörter nicht übereinstimmen werden von CISCON abgewiesen.

Fall 5:

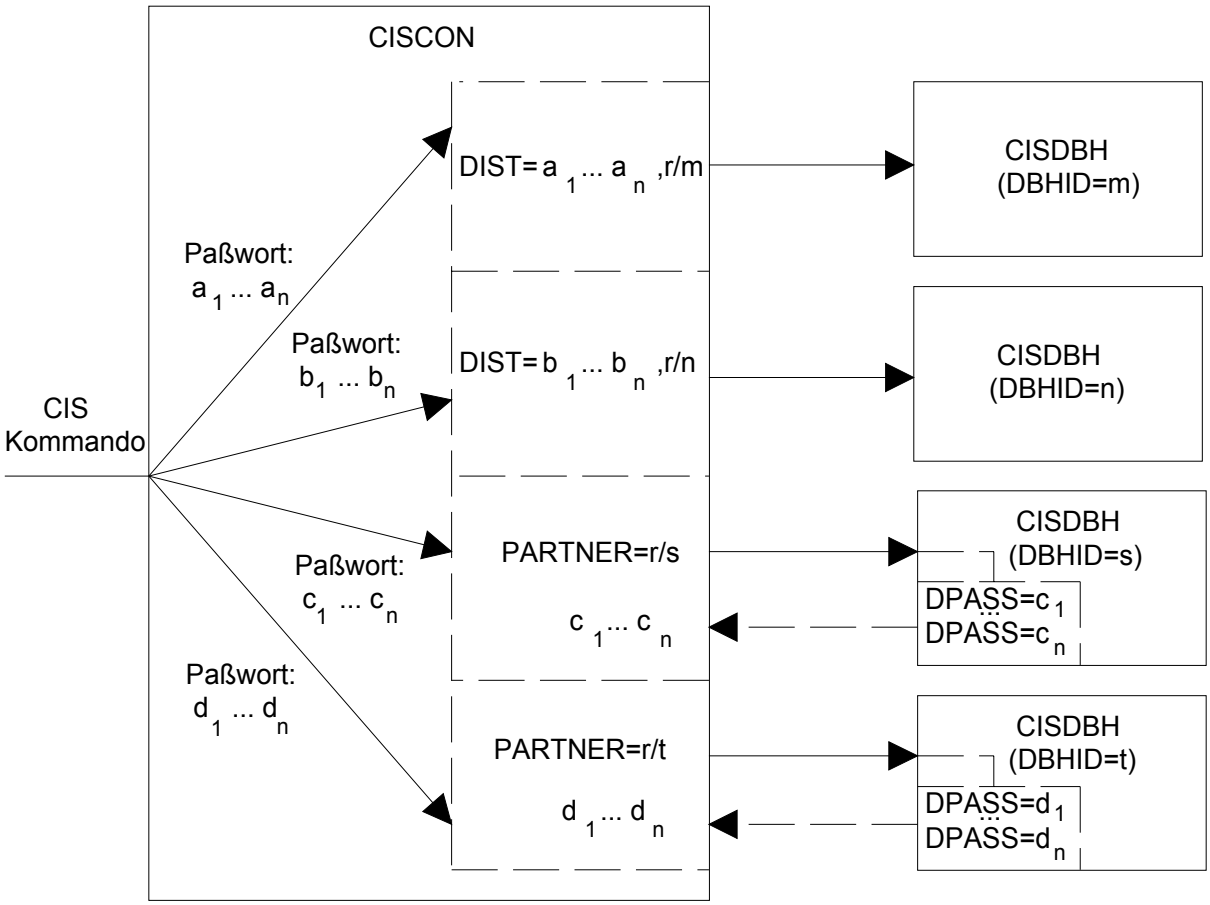


Mehrere PARTNER-Anweisungen teilen CISCON mit, mit welchem CISDBH auf einem bestimmten Rechner gearbeitet werden soll. Damit ist vorerst nur der Rahmen abgesteckt und die Arbeit noch nicht in Einzelheiten auf die CISDBH's verteilt.

Die DPASS-Anweisung teilt dem jeweiligen CISDBH die Paßwörter mit, mit denen die Aufträge gekennzeichnet sein müssen, um bearbeitet werden zu können. Bei der ersten Verbindungsaufnahme von CISCON mit CISDBH werden die Paßwörter an CISCON übergeben. Aufträge mit nicht übereinstimmenden Paßwörten werden von CISCON abgewiesen.

Aus der Sicht von CISCON können die Fälle 4 und 5 gemeinsam auftreten.

Zu Fall 4+5:



4.6 Beispiele von Parametern

In den folgenden Beispielen werden jeweils die Parameter für CISCON und für CISDBH angegeben.

Beispiel 1

Es soll mit CISDBH mit der Kennung A gearbeitet werden. CISDBH ist im gleichen Rechner. Er soll automatisch gestartet werden. CISDBH arbeitet mit einem eingebundenen CISKOOR und beschreibt nur eine AFTER-Image-Datei.

CISCON-Parameter:
```
ENT-JOB=E.CISDBH.A
DBH=*/A
END
```

Datei: `E.CISDBH.A:`

```
/LOGON
/SET-FILE-LINK FILE-NAME=CIS.AIM,LINK-NAME=AIM
/SET-FILE-LINK FILE-NAME=DABEL,LINK-NAME=DB
/START-PROGRAM FROM-FILE=CISDBH
DKOOR=I                 inlinked CISKOOR
DAIM=J                  AIM-Datei
DBIM=N
DTRANS=N
DBHID=A
DEND=1
DPOOL=06                Memory-Pool-Größe: 6x64K=384KB
END
/LOGOFF
```

Beispiel 2

Es soll mit zwei CISDBH mit den Kennungen A und B gearbeitet werden. CISDBH-A bearbeitet die CIS-Paßworte PASSW1 und PASSW2, CISDBH-B bearbeitet das CIS-Paßwort PASSW3. Die beiden Programme sollen automatisch gestartet werden. Für CISDBH-A und für CISDBH-B sollen jeweils 3 Tasks vorhanden sein.

CISCON-Parameter:

```
ENT-JOB=E.CISDBH.A
PARTNER=*/A
PARTNER=*/B
END
```

Datei E.CISDBH.A

```
/LOGON
/.EXEC START-PROGRAM FROM-FILE=CISDBH
DKOOR=N                          Keine Datensicherung
DAIM=N
DBIM=N
DTRANS=N
DBHID=A
DPOOL=6
DEND=1
DPASS=PASSW1
DPASS=PASSW2
DMAXT=10                         maximal 10 Tasks möglich
DNTASKS=3                        3 Tasks
DENT-JOB=E.CISDBH.A
END
/SKIP-COMMANDS TO-LABEL=.EXEC, IF=JOB-SWITCHES(ON=27)
                                 Wenn CIS-Fehler wird CISDBH wieder geladen
/LOGOFF
```

Datei E.CISDBH.B

```
/LOGON
/.EXEC START-PROGRAM FROM-FILE=CISDBH
DKOOR=N
DAIM=N
DBIM=N
DTRANS=N
DBHID=B
DPOOL=6
DEND=1
DPASS=PASSW3
DMAXT=10                         maximal 10 Tasks möglich
DNTASKS=3                        3 Tasks
DENT-JOB=E.CISDBH.b
END
/SKIP-COMMANDS TO-LABEL=.EXEC,IF=JOB-SW(ON=27)
                                 Wenn CIS-Fehler wird CISDBH wieder geladen
/LOGOFF
```

Beispiel 3

Wie Beispiel 2, jedoch wird die Verteilregel für CISDBH-A in CISCON angegeben.

CISCON-Parameter:

```
ENT-JOB=E.CISDBH.A
DIST=PASSW1,*/A
DIST=PASSW2,*/A
PARTNER=*/B
END
```

Datei E.CISDBH.A

```
/LOGON
/.EXEC START-PROGRAM FROM-FILE=CISDBH
DKOOR=N
DAIM=N
DBIM=N
DTRANS=N
DBHID=A
DPOOL=6
DEND=1
DMAXT=10
DNTASKS=3
DENT-JOB=E.CISDBH.A
END
/SKIP-COMMANDS TO-LABEL=.EXEC,IF=JOB-SW(ON=27)
/LOGOFF
```

Datei E.CISDBH.B

```
/LOGON
/.EXEC START-PROGRAM FROM-FILE=CISDBH
DKOOR=N
DAIM=N
DBIM=N
DTRANS=N
DBHID=B
DPOOL=6
DEND=1
DPASS=PASSW3
DMAXT=10
DNTASKS=3
DENT-JOB=E.CISDBH.B
END
/SKIP-COMMANDS TO-LABEL=.EXEC,IF=JOB-SW(ON=27)
/LOGOFF
```

Beispiel 4

Es soll mit zwei CISDBHs mit der Kennung A gearbeitet werden. Der eine ist im eigenen Rechner, der andere im Rechner HOST2. CISDBH im eigenen Rechner verarbeitet das CIS-Paßwort PASSW1, CISDBH im Rechner HOST2 das CIS-Paßwort PASSW2. Es wird nicht mit automatischem ENTER gearbeitet.

CISCON-Parameter:
```
PARTNER=*/A
PARTNER=HOST2/A
END
```

Enter-Datei im eigenen Rechner:

```
/LOGON
/START-PROGRAM FROM-FILE=CISDBH
DBHID=A
DEND=2                  Beendigungsbedingung = 2
DFROM=0700              von 7 Uhr
DTO=1800                bis 18 Uhr
DPASS=PASSW1
END
/LOGOFF
```

Enter-Datei im Rechner HOST2 (muß auf diesem Rechner gestartet werden):

```
/LOGON
/START-PROGRAM FROM-FILE=CISDBH
DCAM=J
DBHID=A
DEND=2
DFROM=0700
DTO=1800
DPASS=PASSW2
END
/LOGOFF
```

5 CISKOOR

5.1 Funktionsumfang

CISKOOR ist der CIS-Koordinationsprozeß. Er kann auf zwei Arten eingesetzt werden:

- independent (unabhängig)- CISKOOR ist ein eigener Prozeß
- inlinked (eingebunden) - die CISKOOR-Funktionen werden von CIS ausgeführt

Der Koordinationsprozeß (independent oder inlinked) wird benötigt zum:

1. Schreiben der After-Images
2. Schreiben der Before-Images
3. Schreiben der Protokollsätze
4. Rücksetzen von Transaktionen (=Rückspeichern der Before-Images)
5. Verwaltung der Sperrlisten

CISKOOR wird vor dem ersten Einsatz an die jeweiligen Belange angepaßt (vgl. Manual-1: Installieren der Datensicherung) z.B. mit oder ohne After-Images.

Gestartet wird CISKOOR in einem ENTER-Prozeß. CISKOOR kann im sogenannten Multi-Task-Betrieb gefahren werden. Es können mehrere Tasks geladen werden. Diese arbeiten im Verbund und verteilen unter sich die Last vollkommen dynamisch.

Auf einem Rechner können gleichzeitig mehrere Gruppen von CISKOOR-Tasks ablaufen. Zur Unterscheidung erhält jede Gruppe eine 1-Byte-Kennung (sog. KOORID).

CISKOOR kann mit Parametern weitgehend an die eigenen Bedürfnisse angepaßt werden.

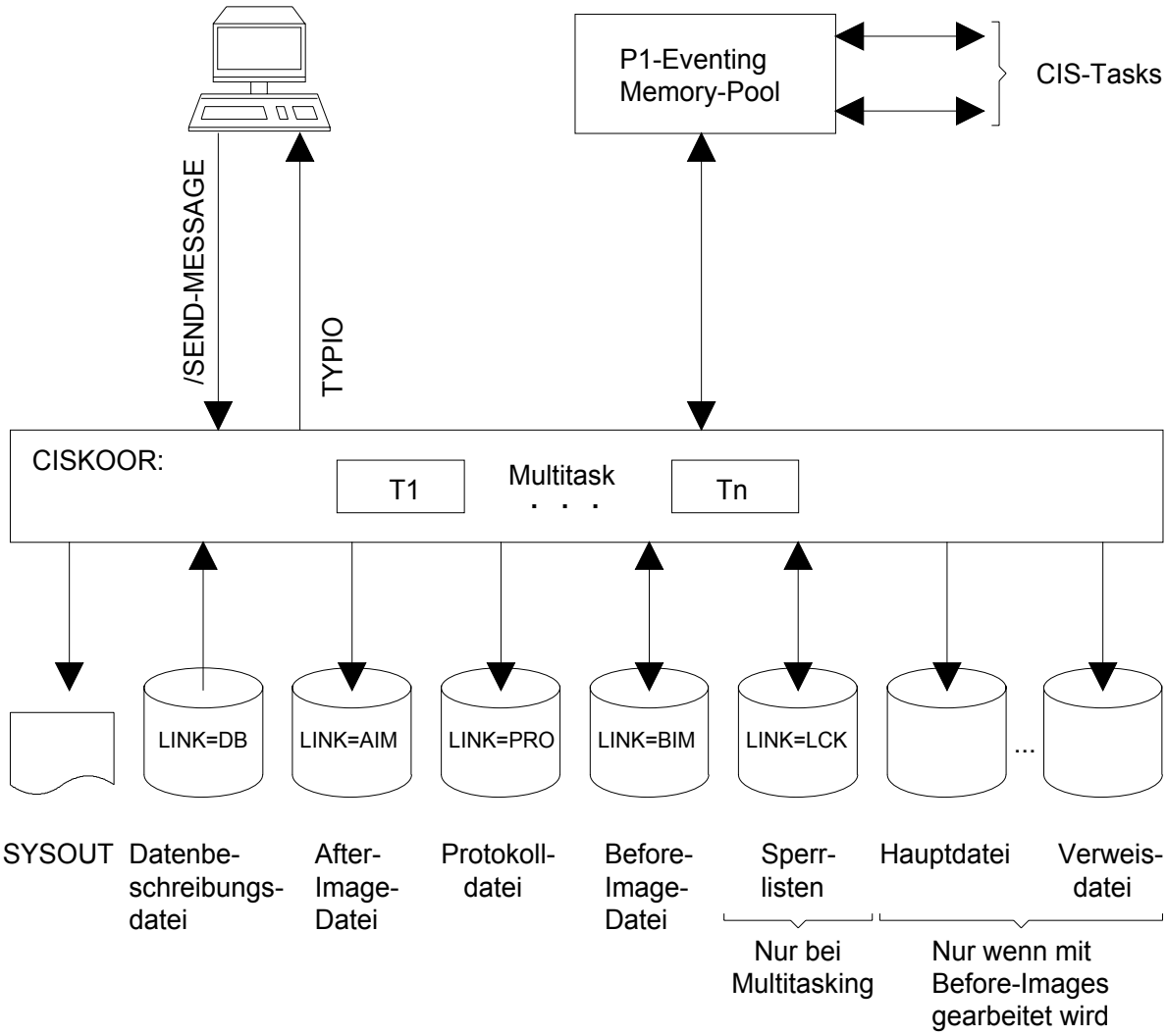
Eingriffe während des Betriebes können über das /SEND-MESSAGE Kommando oder über das CIS-Kommando '$D' vorgenommen werden.

CISKOOR-V12 kann als Partner Programme der CIS Versionen V9, V10, V11 und V12 bedienen. Alle CISKOOR einer Gruppe müssen die gleiche Version haben.

CISKOOR läuft als Transaktionsprozeß, sofern es im Join-Eintrag erlaubt ist.

CISKOOR bedient das "2-Phasen-Commit-Protokoll". Somit kann CIS UTM in allen vorhandenen Varianten bedienen.

5.2 Datenflußplan von CISKOOR



5.3 Dateien

CISKOOR benützt folgende Dateinamen:

Datei	LINK-NAME	FCB-Typ	Standard-Dateiname
Datenbeschreibungsdatei	DB	ISAM	DABEL
After-Image-Datei	AIM	PAM	CIS.AIM
Before-Image-Datei	BIM	PAM	CIS.BIM
Sperrlisten	LCK	PAM	CIS.LCK
Protokolldatei	PRO	ISAM	CIS.PRO
CIS-Dateien (BIM-Routinen)		ISAM	aus BIM-Satz

Bei Verwendung des Standard-Dateinamens muß die jeweilige Datei nicht mit dem LINK-Parameter zugewiesen werden.

Die Datensicherungsdateien werden vor dem allerersten Start von CISKOOR mit dem CREATE-FILE und SET-FILE-LINK Kommando eingerichtet und mit dem Programm CISINIT (vgl. Seite 241) initialisiert. Da die After-Image-Datei und die Protokolldatei von CISKOOR immer fortgeschrieben werden (auch nach einem neuen Start), müssen sie regelmäßig mit dem Programm CISINIT wieder rückgesetzt werden. (Vgl. Dienstprogramm CISRA - Seite 245).

Die Before-Image-Datei wird nur vor dem ersten Start mit CISINIT bearbeitet, und dann nie mehr. Ausnahme ist ein geplanter Kaltstart (vgl. Seite 205).

Sind beim Start von CISKOOR keine Dateien vorhanden, so legt CISKOOR sie mit ihrem Standardnamen an. Die Dateigröße wird mit PRIMARY-ALLOCATION=300 und SECONDARY-ALLOCATION=30 angelegt. Sind die Dateien noch nicht initialisiert, so werden sie von CISKOOR initialisiert. Das Anlegen der Dateien wird nicht ausgeführt wenn der Parameter KFILE=N angegeben wurde.

Beispiel einer ENTER-Datei:

```
/LOGON
/SET-FILE-LINK FILE-NAME=CIS.DABEL,LINK-NAME=DB
/SET-FILE-LINK FILE-NAME=CIS.AIM.T,LINK-NAME=AIM
/SET-FILE-LINK FILE-NAME=CIS.BIM.T,LINK-NAME=BIM
/SET-FILE-LINK FILE-NAME=CIS.LCK.T,LINK-NAME=LCK
/SET-FILE-LINK FILE-NAME=CIS.PRO.T,LINK-NAME=PRO

/START-PROGRAM FROM-FILE=CISKOOR
.                                  ⎫
.                                  ⎬  Parameter
.                                  ⎭
END
/LOGOFF
```

Die LINK-Zuweisungen sind hier erforderlich, da nicht die Standard-Dateinamen verwendet werden.

5.4 Einrichten von CISKOOR

Das ausgelieferte Programm CISKOOR ist auf den Normalfall ausgelegt. In der jeweiligen Freigabemitteilung ist erläutert, wie CISKOOR ausgeliefert wird. Soll CISKOOR anders eingesetzt werden, so ist die Prozedur, D.CIS.DASI.INSTALLIEREN, entsprechend abzuändern (vgl. Manual-1: Installieren Datensicherung). Folgendes kann geändert werden:

- AFTER-IMAGES schreiben?
- VD-AFTER-IMAGES schreiben?
- PROTOKOLL-Sätze schreiben?
- CISKOOR-TYP?
- Beendigungsbedingung?
- Normaler Einsatz von - bis?
- Automatischer Enter für CISKOOR?
- Größe des Memory-Pools?

5.5 Start des CISKOOR-Prozesses

5.5.1 Startprozedur

Folgende ENTER-Prozedur muß definiert werden:

```
/LOGON
/SET-FILE-LINK LINK=DB,F-NAME=Name
/SET-FILE-LINK LINK=AIM,F-NAME=Name
/SET-FILE-LINK LINK=BIM,F-NAME=Name
/SET-FILE-LINK LINK=LCK,F-NAME=Name
/SET-FILE-LINK LINK=PRO,F-NAME=Name
/START-PROGRAM FROM-FILE=CISKOOR
.
.
.
END
/LOGOFF
```

Nur notwendig, wenn die Dateien vorhanden sind und von den Standarddateinamen abweichen.

Parameter

Der Start des Prozesses kann auf 2 Arten erfolgen:

- ENTER-JOB Kommando durch den Systemverwalter oder Operator (z.B. beim Systemstart), bzw. durch den CIS-Benutzer.
- Automatisches ENTER-JOB Kommando in CIS (muß bei der Einrichtung der Datensicherung generiert werden - vgl. Manual-1: Installieren Datensicherung)

5.5.2 Multi-Task-Betrieb

Es können mehrere Tasks im Verbund arbeiten und so die Last dynamisch verteilen. Wenn von CISKOOR die Rede ist, dann ist ein Verbund von mehreren CISKOOR-Tasks gemeint.

Beim Starten werden Parameter zur Definition des Verbundes benützt.

KMAXT=n	legt die maximale Anzahl von Tasks für diesen CISKOOR fest. Der Parameter wird benötigt, weil bestimmte Tabellen anhand der Maximalanzahl von Tasks dimensioniert werden. Ist der Wert größer 1, so muß der Parameter KENT-JOB angegeben werden.
KNTASKS=n	legt die Anzahl Tasks fest, die ab Start von CISKOOR laufen sollen. Der erste Task wird immer manuell (von außen) gestartet. Die weiteren Tasks (bei n > 1) werden von dem ersten CISKOOR gestartet.
KENT-JOB=Parameter	gibt die Parameter für das ENTER-JOB Kommando, das CISKOOR intern absetzt, um weitere Tasks zu starten.

Während des Betriebs kann die Anzahl der Tasks jederzeit zwischen 1 und dem Maximalwert mit dem Kommando $D KNTASKS geändert werden.

Beim Multitask-Betrieb wird die Last dynamisch verteilt. Die Zugriffe auf Sperrlisten, After-Image-Sätze und Before-Image-Sätze sind serialisiert.

5.5.3 Mehrere CISKOOR

Auf einer Anlage können mehrere CISKOOR laufen (ab CIS-Version 11). Unter CISKOOR versteht sich jeweils ein Verbund von mehreren Tasks, die sich die Arbeit teilen (Multitask-Betrieb). Ein solcher CISKOOR-Verbund erhält eine einstellige Kennung mit dem Parameter KOORID.

Nur ein CISKOOR ohne KOORID kann CIS-V9, CIS-V10 und CIS-V11 gleichzeitig mit CIS-V12 bedienen.

Auf einer Anlage können somit bis zu 37 CISKOOR laufen (Ein CISKOOR ohne KOORID, 26 mit KOORID von A bis Z und 10 mit KOORID von 0 bis 9).

Ein CIS kann jeweils nur mit einem dieser CISKOOR arbeiten.

Eine bestimmte CIS-Datei darf nur in einem CISKOOR behandelt werden. Dies gilt für die Lebensdauer dieses CISKOOR. Die Prüfung findet beim Schreiben des ersten Before-Image-Satzes für diese Datei und diesen CIS-Task statt.

Die einzelnen CISKOOR arbeiten völlig getrennt. Jeder hat seine Datensicherungsdateien und läuft vollkommen parallel zu anderen CISKOOR. Es gibt eine wirkliche Lastaufteilung ohne Serialisierungseffekte. Datensicherungsdateien von einem CISKOOR (Verbund einzelner Tasks) dürfen nicht von einem anderen CISKOOR benützt werden. Um dies prüfen zu können wird die CISKOOR-Kennung in jede Datensicherungsdatei beim ersten Eröffnen durch CISKOOR geschrieben. Ab diesem Zeitpunkt kann diese Datei nur noch von dieser CISKOOR-Kennung benützt werden. Das Programm CISINIT löscht die CISKOOR-Kennung der Datei.

Anwendungsbeispiel von verschiedenen CISKOOR:

Die Anwendungen, die Daten auf dem Pubset "A" bearbeiten, werden von CISKOOR "A", die Anwendungen, die Daten auf Pubset "B" bearbeiten, von CISKOOR "B" bedient.

Werden Anwendungen mit ihrem Pubset ausgelagert, so wird auch der entsprechende CISKOOR ausgelagert.

5.5.4 Kaltstart, Warmstart und fehlertoleranter Warmstart

Definitionen:

Kaltstart: CISKOOR ignoriert eventuell vorhandene Before-Images, auch wenn diese einen vorhergehenden Crash überlebt haben.

Warmstart: CISKOOR liest die Before-Images und setzt die offenen Transaktionen zurück. Sind Transaktionen im Zustand PTC (Prepare To Commit), so bleiben deren Sätze gesperrt. Diese Transaktionen bleiben offen, bis ein Signal ihres Koordinators (z.B. UTM-D) kommt.

Fehlertoleranter Warmstart:

Wie Warmstart. Kann ein Before-Image Satz nicht verarbeitet werden, so wird er ignoriert. Am Ende der Verarbeitung wird für jede Datei, bei der Sätze nicht verarbeitet werden konnten, der Dateiname und der Fehlercode protokolliert.

CISKOOR führt standardmäßig einen Warmstart aus.

Soll ein Kaltstart ausgeführt werden, so gibt es folgende Möglichkeiten:

- Parameter KSTART=C angeben.
- CISINIT auf BIM-Datei vor dem CISKOOR-Start.

Achtung!

Bei einem Kaltstart geht folgendes verloren:

- Before-Images Die Datenkonsistenz ist gefährdet.
- PTC-Zustand Die Konsistenz der verteilten Transaktion ist gefährdet.
- Status-Infos Der Koordinator (z.B. UTM) erhält für seinen Warmstart falsche Informationen.

5.6 Verbindung CIS - CISKOOR

CISKOOR ohne KOORID

Die Verbindung von CIS zu CISKOOR funktioniert wie in den CIS-Versionen 9 bis 10. CIS findet CISKOOR auf Grund der Daten, die in CISVARI eingetragen sind. CISKOOR kann CIS der Versionen V9, V10, V11 und V12 gleichzeitig bedienen.

CISKOOR mit KOORID

CIS wird mit folgendem SET-FILE-LINK Kommando mitgeteilt, mit welchem CISKOOR es Verbindung aufnehmen soll.

```
/SET-FILE-LINK FILE-NAME=CISKOOR.i,LINK-NAME=CISKOOR
```

`i` KOORID

Das Anmelden von CIS-Tasks bei CISKOOR kann mit folgendem Kommando gesperrt werden.

$D KENA L (ENA steht für Enable Communication).

Somit kommt kein neuer Partner mehr hinzu. Die noch bestehenden Partner können mit $D KINFT abgefragt werden.

5.7 Beenden von CISKOOR

Normales Beenden von CISKOOR.
- CISKOOR wird auf Beendigungsbedingung 1 gesetzt.
- Alle Partner melden sich bei CISKOOR ab.
- Dann beendet sich CISKOOR.

Abbruch von CISKOOR.

`/CANCEL-JOB JOB-ID=TSN(tsn)`

`/BCLOSE CISKOOR` (nur Konsole) Wenn Parameter KDCAM=J, wurde CISKOOR als DCAM-Anwendung angemeldet.

`/BCEND` (nur Konsole) Wenn Parameter KDCAM=J, wurde CISKOOR als DCAM-Anwendung angemeldet.

Wird CISKOOR mit diesen Modi beendet, wird in CISKOOR die Ende-Routine aufgerufen. Diese setzt alle offenen Transaktionen zurück und schließt die Dateien.

5.8 Benützte Schalter

Die Prozeßschalter 25 und 28 werden von CISKOOR benützt (CISKOOR setzt beim Start die beiden Prozeßschalter auf OFF).

25 = ON nach einer Fehlermeldung während des CISKOOR-Laufs.

28 = ON wenn die Before-Image-Datei nach dem CISKOOR-Lauf noch Before-Images enthält.

5.9 Memory-Pool

5.9.1 Allgemeines

Die Kommunikation zwischen CIS (alle Programme, die das Modul CISI benützen, also auch CISDBH) und CISKOOR bedient sich zweier Möglichkeiten des Betriebssystems:

1. P1-Eventing — Zum Übertragen von Signalen, also zum Melden, daß eine Nachricht vorhanden ist.
2. Common-Memory-Pool — Zum Übergeben der Daten.

Die Kommunikation läuft automatisch, der Benutzer muß lediglich die Größe des Common-Memory-Pools festlegen. Die folgenden Erläuterungen sollen helfen, die Größe des Common-Memory-Pools richtig zu bestimmen.

5.9.2 Benützung des Memory-Pools

CISKOOR richtet den Memory-Pool ein und legt auch seine Größe fest. Diese kann innerhalb eines Laufes nicht mehr verändert werden.

Der erste Task von CISKOOR richtet den Memory-Pool ein. Alle folgenden Tasks schließen sich an den Pool an. Der Pool existiert so lange im System wie irgendein Task damit arbeitet. Es kann also vorkommen, daß alle CISKOOR-Tasks beendet werden, der Pool aber beibehalten wird, weil noch ein CIS-Task aktiv ist. Wird jetzt ein CISKOOR-Task gestartet, so meldet er den Fehler "alter Memory-Pool". Damit CISKOOR in diesem Fall wieder gestartet werden kann, müssen zuvor alle CIS-Programme, die mit CISKOOR gearbeitet haben, beendet werden.

Nach dem Einrichten des Memory-Pools belegt CISKOOR folgenden Speicher:

4 KB

CISKOOR-Tabelle (KMAXT * 8)

Partner-Tabelle (KNCOMM * (48 + KMAXT * 8))

Transaktionstabelle (KNTRANS * 96)

Status-Tabelle (KNSTA * 56)

Nimmt man die Default-Werte für die einzelnen Parameter, jedoch den Wert 10 für KMAXT an, so belegt CISKOOR

4K+(10*8)+(100*(48+10*8))+(200*96)+(400*56)=ca. 58 KB.

Jeder CIS-Task, der mit CISKOOR Verbindung aufnimmt, belegt einen Teil des Memory-Pools. Mit Hilfe dieses Bereiches werden Informationen zwischen CIS-Task und CISKOOR ausgetauscht. Der Bereich muß so groß sein, daß der gößtmögliche After-Image- oder Before-Image-Satz hineinpaßt. Dies entspricht dem größtmöglichen HD-Satz wie in CISVARI + E4 angegeben. Zusätzlich wird Platz für diverse Verwaltungsinformationen benötigt wie z.B. Dateiname usw. Diese Informationen belegen ca. 200 Bytes. Die Teile im Memory-Pool werden nur in Blöcken von 4K vergeben.

5.9.3 Berechnen der Memory-Pool Größe

Als erstes muß der Platz für einen CIS-Task errechnet werden. Nach den vorhergehenden Erläuterungen ergibt sich folgende Tabelle:

Max. Größe HD-Satz CISVARI + E4 (Vielfaches von 2 KB)	Platzbedarf für einen CIS-Task
01, 02, 03	8 K
04, 05	12 K
06, 07	16 K
08, 09	20 K
10, 11	24 K
12, 13	28 K
14, 15	32 K
16	36 K

Die Größe des Memory-Pools errechnet sich nach folgender Formel:

Größe Memory-Pool = 58 + n * c (in KB)

n = Anzahl gleichzeitig laufender CIS-Tasks

c = Größe des Platzes für einen CIS-Task

Die Größe des Memory-Pools wird auf NXS-Anlagen auf ein Vielfaches von 64 KB, auf XS31-Anlagen auf ein Vielfaches von 1 MB aufgerundet.

5.9.4 Anzahl der CIS-Tasks

Maximale Anzahl gleichzeitiger CIS-Tasks in Abhängigkeit der Memory-Pool Größe und der maximalen HD-Satzgröße:

Poolgröße		HD-Satzgröße (Vielfaches von 2 KB)							
à 64 K	KB/MB	1 2 3	4 5	6 7	8 9	10 11	12 13	14 15	16
1	64 KB	0	0	0	0	0	0	0	0
2	128 KB	8	5	4	3	2	2	2	1
4	256 KB	24	16	12	9	8	7	6	5
8	512 KB	56	37	28	22	18	16	14	12
16	1 MB	120	80	60	48	40	34	30	26
32	2 MB	248	165	124	99	82	71	62	55
64	4 MB	504	336	252	201	168	144	126	112

5.10 Bemerkungen zum 2-Phasen-Commit-Protokoll

5.10.1 Allgemeines

Das 2-Phasen-Commit-Protokoll kommt bei einer verteilten Transaktion zur Anwendung. Die verteilte Transaktion kann als geschachtelte Transaktion angesehen werden. Die Wurzel des Transaktionsbaums nennen wir "Koordinator" (engl. coordinator), die Subtransaktionen "Agenten" (engl. resource managers oder participants).

Wenn eine verteilte Transaktion korrekt beendet wird, dann müssen auch alle Subtransaktionen korrekt beendet worden sein. Im anderen Fall müssen alle Subtransaktionen zurückgesetzt werden. Wird dies nicht gewährleistet, so ist die verteilte Transaktion nicht mehr "atomar".

Eine lokale Subtransaktion kann selbst auch wieder eine geschachtelte Transaktion sein.

Folgende grundlegende Idee liegt dem 2-Phasen-Commit-Protokoll zugrunde:

> Für alle beteiligten Subtransaktionen wird bezüglich des Beendens oder Rücksetzens eine einzige Entscheidung getroffen. Es findet eine Art Abstimmung (engl. vote) statt. Die Transaktion kann nur beendet werden wenn dies einstimmig beschlossen wird.

Der Koordinator schreibt ausfallsichere Logsätze. Kommt es zu einem Crash, erkennt er an Hand dieser Sätze, an welchem Punkt er die Fortsetzung des Protokolls wieder aufnehmen muß.

5.10.2 Ablauf des Protokolls

1. Phase 1

1.1 Der Koordinator schreibt einen "Prepare"-Logsatz. Dieser enthält alle betroffenen Subtransaktionskennungen.

1.2 Der Koordinator sendet an alle Agenten eine "Prepare"-Nachricht.

1.3 Alle Agenten werden damit in einen Zustand versetzt, in dem sie jeweils die entsprechende Subtransaktion beenden oder rücksetzen können. Die Agenten gehen ausfallsicher in diesen Zustand und können auch nach einem Crash ihre Subtransaktion vor- oder rücksetzen.

1.4 Jeder Agent meldet dem Koordinator, ob er erfolgreich war oder nicht.

1.5 Der Koordinator wartet auf alle Antworten der Agenten. Er wertet diese aus und entscheidet, was mit der Transaktion geschehen soll:

> Wenn alle Subtransaktionen erfolgreich waren, wird die Transaktion beendet (engl. commit).
>
> War mindestens eine Subtransaktion nicht erfolgreich, so wird die Transaktion rückgesetzt (engl. abort).

2. Phase 2

2.1 Der Koordinator schreibt je nach vorhergehender Entscheidung einen "global commit" oder einen "global abort" Logsatz.

2.2 Die Entscheidung des Koordinators wird als Nachricht (beenden oder rücksetzen) an alle Agenten geschickt.

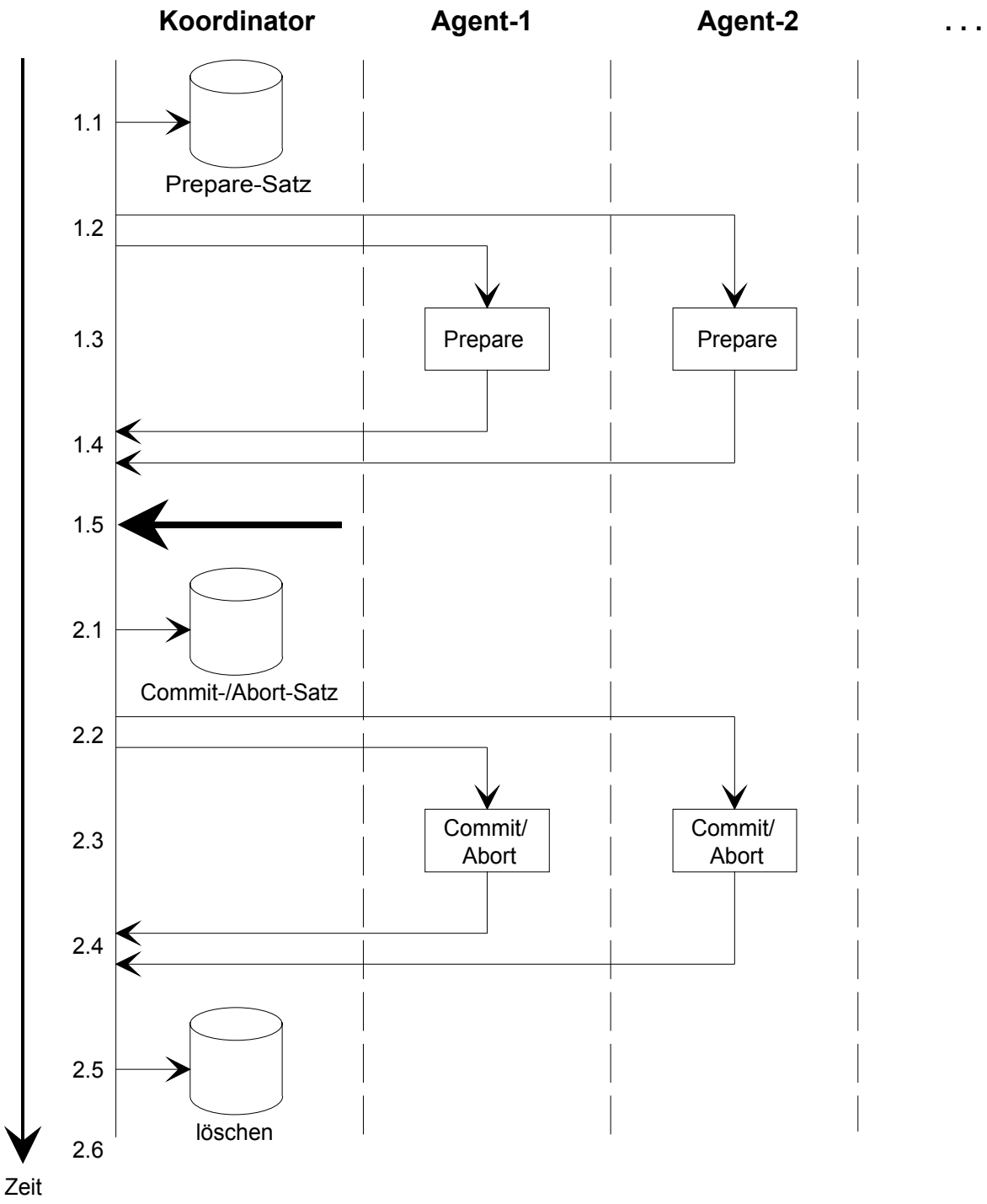
2.3 Die Agenten beenden ihre jeweilige Subtransaktion oder setzen sie zurück.

2.4 Die Agenten senden eine Bestätigung über Beendigung bzw. Rücksetzung an den Koordinator.

2.5 Hat der Koordinator die Antwort aller Agenten erhalten, so löscht er seine Logsätze.

2.6 Die Transaktion ist abgeschlossen.

Darstellung des Ablaufs



Die Punkte entsprechen den Erläuterungen auf den vorhergehenden Seiten: 1.x = Phase 1
2.x = Phase 2

5.11 Bemerkungen zum Warmstart

Es wird kurz erläutert, wie CIS als Datenhaltungssystem im Falle eines Warmstarts reagiert.

5.11.1 Offene Transaktion

Wird eine offene Transaktion gefunden, so wird sie zurückgesetzt. War es eine Transaktion der Betriebsart S oder wurde sie bei den anderen Betriebsarten mit TR,AS gestartet , so wird sie in der Statusdatei als zurückgesetzt vermerkt.

Es kann jederzeit, auch nach einem eventuellen erneuten Crash, der Zustand dieser Transaktion abgefragt werden (TR,SA-Kommando). Dieses Kommando muß vom Koordinator der verteilten Transaktion gegeben werden. Der Koordinator muß auch dafür sorgen, daß der Eintrag in der Statusdatei wieder gelöscht wird (TR,SL-Kommando).

Bei der Betriebsart S erfolgen diese Arbeiten in CISCON in Zusammenarbeit mit UTM.

5.11.2 Transaktion im Zustand PTC

CIS darf diese Transaktion nicht eigenmächtig bearbeiten, sondern muß warten, bis der Koordinator ein diesbezügliches Kommando (TR,WE oder TR,WR) übermittelt.

Die ursprüngliche Transaktionskennung wird in die Transaktionstabelle eingetragen, die angefaßten Sätze (Sätze, für die Before-Images vorhanden sind) werden gesperrt. Nach TR,WE oder TR,WR ist die Transaktion ausgetragen und die Sperren sind aufgelöst.

5.11.3 Manuelle Eingriffe

Manuelle Eingriffe sind über CISKOOR möglich. Es muß jedoch klar sein, daß nur eine Subtransaktion bearbeitet wird, die gesamte ursprüngliche Transaktion kann nicht berücksichtigt werden, da der Koordinator ausgeschaltet wird.

5.11.4 Stati

Stati können mit $D KINFS abgefragt werden und mit $D KCANS gelöscht werden.

5.11.5 PTC Zustände

PTC-Zustände können mit $D KINFP abgefragt werden und mit $D KPEND beendet oder mit $D KPRESET rückgesetzt werden.

5.12 Besonderheiten bei Inlinked CISKOOR

Allgemeines

Es kann auch ein inlinked CISKOOR benützt werden. In diesem Fall sind die Routinen von CISKOOR in CIS eingebunden.

Jedes CIS-Programm hat seinen eigenen CISKOOR. Somit gibt es für jedes CIS-Programm die Datensicherungsdateien.

Dieser Modus ist natürlich wesentlich performanter als der independent Betrieb. Aber um sicherzustellen, daß nicht mehrere CIS-Tasks gleichzeitig in der gleichen Datenbank ändern, werden im Falle des inlinked CISKOOR die Dateien mit SHARUPD=NO und OPEN=INOUT eröffnet; also exklusiv. Es kann kein anderes Programm auf diese Dateien zugreifen.

Einrichten und starten

Zum Einrichten des inlinked CISKOOR muß das Byte CISVARI+X'9' den Wert C'I' enthalten. Damit wird CISKOOR von CIS aufgerufen. Mit dem Anpassen des Moduls CISVARI in CIS, und mit dem Binden von CIS ist CISKOOR automatisch auch angepaßt.

Bedienung während des Betriebs

Die Bedienung ist ausschließlich über das $D-Kommando möglich.

Beenden

Der inlinked CISKOOR wird automatisch mit dem CIS-Programm beendet.

Dateinamen

Wie bei independent CISKOOR (vgl. Seite 201).

5.13 Parameter für CISKOOR

Übersicht:

Parameter	Bedeutung	Default-wert
`*......`	Kommentar	
`CISVARI+d={C'c...c'/ X'xx..xx'}`	Korrektur von CISVARI (variable Parameter)	
`END[_Parameter]`	Ende der Parameter	
`KAIM=J/Y/N`	Verwendung der AIM-Datei	J
`KBIM=J/Y/N`	Verwendung der BIM-Datei	J
`KCANTSN=J/Y/N`	CANCEL-JOB für noch vorhandene Partner eines vorhergehenden Laufs	N
`KCHKBIM=J/Y/N`	Prüfen ob BIM noch vorhanden.	N
`KDCAM=J/Y/N`	Anmeldung von CISKOOR bei DCAM als Anwendung	N
`KDIAG=J/Y/N`	Diagnose einschalten	N
`KEND=e`	Setzen der Beendigungsbedingung	2
`KENT-JOB=Parameter`	Setzen der Parameter zum Starten weiterer CISKOOR-Tasks	Spaces
`KERR=cccc,E/D/T/K`	Reaktion auf Fehlercode	
`KFILE=J/Y/N`	File Kommando für Datensicherungsdateien	J
`KFROM=hhmm`	Betriebszeit: Setzen der Startzeit	0700
`KLOCKP=n`	Größe des Sperr-Pools in KB	1000
`KLOG=J/Y/N`	Verwendung Protokolldatei	N
`KMAXT=n`	Maximale Anzahl der Tasks für CISKOOR	1
`KNCOMM=n`	Anzahl Einträge in Partnertabelle	100
`KNFCB=n`	Anzahl FCB´s bei Restart/Rollback	9999
`KNLOCK=n`	Anzahl Einträge im Sperrverzeichnis	512

Parameter	Bedeutung	Default-wert
KNSTA=n	Anzahl Einträge in der Statustabelle	400
KNTASKS=n	Anzahl Tasks für CISKOOR	1
KNTRANS=n	Anzahl Einträge in der Transaktionstabelle	200
KOORID=i	Identifikation des CISKOOR festlegen	Space
KPETA=WAIT/END/RESET	Peta (PTC) - Bearbeitung bei Warmstart	WAIT
KPOOL=n	Größe des Memory-Pools	16
KPOOLLNK=J/Y/N	POOLLNK bei Warmstart auswerten	J
KSTAT=J/Y/N	Ausgabe Statistikwerte nach Programmende	N
KSTART=C/W/L	Setzen Kalt-/ Warm-/ fehlertoleranter Warmstart	W
KTIMEOLD=J/Y/N	Time-Liste einschalten (SYSLST01)	N
KTIMEON=J/Y/N	Time-Liste einschalten (Datei)	N
KTO=hhmm	Betriebszeit: Setzen Endezeit	1800
KTRANS=J/Y/N	Arbeiten mit Transaktionen	J
KTRON=J/Y/N	Einschalten Trace-Liste	N
SYSDTA=Name	Umweisen von SYSDTA	

Jeder Parameter kann einen Kommentar enthalten. Dieser muß vom Parameter durch mindestens ein Space getrennt sein.

```
*...
```

Kommentar: Es können beliebig viele Kommentare vorhanden sein.

```
CISVARI+d={C'c...c '/X'xx..xx'}
```

Korrektur von CISVARI (variable Parameter für die Ablaufumgebung von CIS)

d	Distanz CISVARI	(sedezimal anzugeben, maximal 4-stellig)
c...c	Text für CISVARI	
xx...xx	Sedezimaler Text für CISVARI (immer gerade Zahl von Zeichen)	

Es können mehrere CISVARI-Parameter angegeben werden.

```
END[_Parameter]
```

Ende der Parameter

Bei Angabe von Parametern im Kommando, werden diese zum Umweisen von SYSDTA benützt. (Es wird intern das Kommando '/ASSIGN-SYSDTA TO-FILE=Parameter' abgesetzt).

```
KAIM=J/Y/N
```

Verwendung der After-Images

J/Y	After-Images
N	Keine After-Images

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird J angenommen.

```
KBIM=J/Y/N
```

Verwendung Before-Images für Datensicherung

J/Y Before-Images
N Keine Before-Images

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird J angenommen.

```
KCANTSN=J/Y/N
```

CANCEL-JOB-Kommandos für alle TSN

J/Y Ja
N Nein

CANCEL-JOB-Kommandos für alle TSN eines vorhergehenden Laufs, bei dem CISKOOR sich beendet hat.

```
KCHKBIM=J/Y/N
```

Prüfen ob BIM noch vorhanden.

J/Y Ja
N Nein

Dieser Parameter dient der Überprüfung ob noch Before-Images vorhanden sind. CISKOOR setzt, falls Before-Images vorhanden sind, den Prozeßschalter 28 und beendet sich ohne weitere Aktionen durchzuführen.

Wichtig: KOORID wird ausgewertet, andere Parameter werden nicht berücksichtigt.

```
KDCAM=J/Y/N
```

Anmeldung von CISKOOR bei DCAM

J/Y Ja
N Nein

CISKOOR meldet sich mit dem Namen 'CISKOOR' bei DCAM als Anwendung an. Mit den BS2000-Kommandos '/BCEND' bzw. '/BCLOSE' läßt sich CISKOOR beenden.

```
KDIAG=J/Y/N
```

Einschalten Diagnose.

J/Y Ja
N Nein

Wenn die Diagnose eingeschaltet ist, werden Before-Image-Sätze und Sperraufrufe auch in die After-Image-Datei geschrieben. SP04-Meldungen werden im folgenden Format von CISKOOR auf SYSOUT protokolliert

```
SP04 TRY : host/appli/tid -x M=m U=user
SP04 LOCK: host/appli/tid -x M=m U=user
SP04 FILE: :cat:$uid.hdname
SP04 K-t : xxx
```

x Kennung L = Liste, K = Key
m Modus (N,U,S)
user User (nur Modus = S)
t Typ C = Character, H = Hexa.
xxx (erster) schon gesperrter Key

```
KEND=e
```

Setzen der Beendigungsbedingung von CISKOOR.

e Beendigungsbedingung (1, 2, 3)

1 = CISKOOR wird nach Abmeldung des letzten Partners beendet.

2 = CISKOOR kann während der Betriebszeit (siehe KFROM- und KTO-Parameter) nicht beendet werden. Außerhalb dieser Zeit geht CISKOOR automatisch auf die Beendigungsbedingung 1.

3 = CISKOOR kann nicht beendet werden. Die Beendigungsbedingung muß, falls CISKOOR beendet werden soll, mit

/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))],MESSAGE=KEND e

oder mit $D KEND auf 1 oder 2 herabgesetzt werden.

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird der Wert 2 angenommen.

```
KENT-JOB=Parameter
```

Parameter zum Starten weiterer CISKOOR-Tasks (vgl. KNTASKS=n - Seite 224).

`Parameter` Parameter für ENT-JOB-Kommando (maximal 200 Bytes).
Vgl. BS2000-Kommando /ENTER-JOB: Der Parameter wird ab dem Operanden "Dateiname" angegeben. (Es wird also /ENTER-JOB FROM-FILE= davor gesetzt).

Dieser Parameter darf nur einmal angegeben werden. Ist er nicht vorhanden, so kann die Anzahl Tasks nicht per CISKOOR-Kommando erhöht werden, sondern durch manuelles Starten der Tasks.

```
KERR=cccc,E/D/T/K
```

Parameter, um im Fehlerfall gezielt reagieren zu können.

`cccc` Fehlercode (Codierte Meldung von CISDBH).

E Bei Fehler cccc wird ein Dump ausgegeben. CISDBH läuft normal weiter. Bei Mehrtaskbetrieb ist nur der Task betroffen, in dem der Fehler erkannt wurde.

D Bei Fehler cccc wird ein Dump ausgegeben. CISDBH läuft normal weiter. Bei Mehrtaskbetrieb sind alle Tasks betroffen.

T Bei Fehler cccc wird ein Dump ausgegeben und CISDBH wird beendet. Bei Mehrtaskbetrieb werden alle anderen Tasks gleich behandelt.

K Bei Fehler cccc werden alle CISDBH-Tasks mit Dump beendet. Alle Partner werden beim nächsten Aufruf von CISDBH ebenfalls mit Dump beendet.

Dieser Parameter kann bis zu 32 mal angegeben werden.

```
KFILE=J/Y/N
```

Anlage der Datensicherungsdateien mit dem CREATE-FILE Kommando.

J/Y Dateien anlegen mit PRIMARY-ALLOCATION=300 und SECONDARY-ALLOCATION=30
N Dateien nicht anlegen

Dieser Parameter darf nur einmal angegeben werden. Ist er nicht vorhanden, so wird J angenommen.

```
KFROM=hhmm
```

Betriebszeit: Setzen der Startzeit, ab der CISKOOR bei Beendigungsbedingung 2 nicht beendet werden kann.

hhmm Zeit in der Form: Stunde, Minuten

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird 0700 angenommen.

```
KLOCKP=n
```

Festlegen der Größe des Sperr-Pools in KB.

Dieser Pool enthält die Sperrlisten. Wenn er voll ist, werden die Sperrlisten in EAM (monotask) oder PAM (multitask) ausgelagert.

Dieser Parameter darf nur einmal angegeben werden. Ist er nicht vorhanden, so werden 1000KB reserviert.

```
KLOG=J/Y/N
```

Verwendung Protokolldatei.

J/Y Protokolldatei
N Keine Protokolldatei

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird N angenommen.

```
KMAXT=n
```

Maximale Anzahl Tasks für CISKOOR. Im laufenden Betrieb kann die Anzahl Tasks jederzeit zwischen 1 und dem maximalen Wert geändert werden.

n Max. Anzahl Tasks für CISKOOR

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird der Wert 1 angenommen (Kein Multitask-Betrieb).

```
KNCOMM=n
```

Anzahl der Partner für CISKOOR.

n Max. Anzahl der Partner, die gleichzeitig mit CISKOOR Verbindung aufnehmen können.

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird der Wert 100 angenommen.

```
KNFCB=n
```

Anzahl FCB's bei Restart/Rollback.

n Es können maximal n FCB's (n verschiedene Dateien) beim Restart/Rollback verarbeitet werden.

Dieser Parameter darf nur einmal angegeben werden. Ist er nicht vorhanden, so wird der Wert 999 angenommen.

```
KNLOCK=n
```

n Anzahl Einträge im Sperrverzeichnis.

Pro Transaktion(die Sätze sperrt) und Datei wird ein Eintrag im Sperrverzeichnis belegt. Werden von einer Transaktion für die gleiche Datei mehrmals Sätze gesperrt, so wird nur ein Eintrag belegt.

Dieser Parameter darf nur einmal angegeben werden. Wird der Parameter nicht angegeben, so wird 512 angenommen.

```
KNSTA=n
```

n Anzahl Einträge in der Statustabelle.

Diese Tabelle enthält Informationen über Stati von rückgesetzten Transaktionen und Informationen über Transaktionen, die zum Zeitpunkt des Warmstarts im Zustand Peta waren. (Peta = Provisorisches Ende der Transaktion, auch PTC = Prepare To Commit genannt).

Dieser Parameter darf nur einmal angegeben werden. Wird der Parameter nicht angegeben, so wird 400 angenommen.

```
KNTASKS=n
```

n Anzahl Tasks für CISKOOR.

Anzahl der Tasks für CISKOOR nach dem Start. Wenn n > 1 muß der Parameter 'KENT-JOB=Parameter' (vgl Seite 221) vorhanden sein.

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird der Wert 1 angenommen.

```
KNTRANS=n
```

Anzahl der Transaktionen CISKOOR.

n Max. Anzahl der Transaktionen, die CISKOOR verwalten kann

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird der Wert 200 angenommen.

```
KOORID=i
```

Identifikation des CISKOOR festlegen.

i 1-Byte-CISKOOR-Kennung (A - Z, 0 - 9) oder NO (= keine CISKOOR-Kennung).

Dieser Parameter darf nur einmal angegeben werden. Ist er nicht vorhanden, oder wird KOORID=NO angegeben, gibt es keine CISKOOR-Kennung.

```
KPETA=WAIT/END/RESET
```

Bearbeitung der Transaktionen im Peta-Zustand bei Warmstart. (Peta = Provisorisches Ende der Transaktion, auch PTC = Prepare To Commit genannt).

WAIT Warten bis UTM meldet, was zu geschehen hat.

END Beenden (kann zum Konflikt mit UTM führen).

RESET Rücksetzen (kann zum Konflikt mit UTM führen).

Ist der Parameter nicht angegeben, so wird der Wert 'WAIT' angenommen. Für weitere Informationen siehe 'Verteilte-Transaktionsverarbeitung mit UTM'.

```
KPOOL=n
```

Festlegen der Memory-Pool-Größe.

n Größe des Memory-Pools in Vielfachem von 64 K.

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird der Wert 16 angenommen.

```
KPOOLLNK=J/Y/N
```

POOLLNK bei Warmstart auswerten

J/Y Ja
N Nein

Wenn KPOOLLNK=N, wird beim Warmstart der POOLLNK-Eintrag im FCB ignoriert. (Z.B. wenn ISAM-Pool nicht geladen werden kann.)

```
KSTAT=J/Y/N
```

Ausgabe einer Statistik bei Programmende. Die Statistik wird vom CIS-Team ausgewertet.

J/Y Statistik ausgeben
N Keine Statistik ausgeben

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird N angenommen.

```
KSTART=C/W/L
```

Festlegung Kalt- oder Warmstart. Bei Kaltstart werden alle Before-Image-Sätze gelöscht.

C Kaltstart
W Warmstart
L Fehlertoleranter Warmstart ('Lauwarmstart')

Dieser Parameter darf nur einmal angegeben werden. Ist er nicht vorhanden, so wird W angenommen. (Siehe auch Seite 205).

```
KTIMEOLD=J/Y/N
```

TIME-Liste einschalten (SYSLST01).

J/Y Ja
N Nein

Protokollierung der CISKOOR-Aufrufe auf SYSLST01.

```
KTIMEON=J/Y/N
```

Time-Liste einschalten (Datei).

J/Y Ja
N Nein

Protokoll der CISKOOR-Aufrufe in die Datei CIS.TIME.tttt (tttt=TSN von CISKOOR). Die Datei kann vom CIS-Team ausgewertet werden.

```
KTO=hhmm
```

Betriebszeit: Setzen der Endezeit, bis zu der der CISKOOR nicht beendet werden kann, wenn die Beendigungsbedingung 2 ist.

hhmm Zeit in der Form: Stunde, Minuten

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, wird 1800 angenommen.

```
KTRANS=J/Y/N
```

Angabe, ob mit Transaktionen gearbeitet wird.

J/Y Transaktionen
N Keine Transaktionen

Dieser Parameter darf höchstens einmal angegeben werden. Ist er nicht vorhanden, so wird J angenommen.

```
KTRON=J/Y/N
```

Einschalten der Trace-Liste.

J/Y Ja
N Nein

```
SYSDTA=Name
```

Zuweisen von SYSDTA. Der nächste Parameter wird von dieser Datei gelesen.

Name Name der Datei von der per RDATA zu lesen ist. (SYSDTA-Datei).

Dieser Parameter kann öfters angegeben werden. Statt "Name" kann auch *PRIMARY oder *SYSCMD angegeben werden.

Es wird intern das Kommando

```
/ASSIGN-SYSDTA TO-FILE=Name
```

abgesetzt.

5.14 Bedienung während des Betriebs

Die Bedienung erfolgt über das BS2000-Kommando '/SEND-MESSAGE'. Es besteht die Möglichkeit von einem gerade arbeitenden CIS-Programm mit dem CIS-Kommando '$D' die SEND-MESSAGE Schnittstelle zu simulieren.

Übersicht über die Kommandos

Kommando	Bedeutung
/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))], MESSAGE=KENA_x $D_KENA_x[,DB.xxxxxx]	Behandlung der ENA-Comm (enable communication)
/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))], MESSAGE=KEND_b $D_KEND_b[,DB.xxxxxx]	Setzen der Beendigungs-bedingung des CISKOOR
/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))], MESSAGE=KINFB[_p] $D_KINFB[_p][,DB.xxxxxx]	Information über Before-Images
/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))], MESSAGE=KINFC[_p] $D_KINFC[_p][,DB.xxxxxx]	Information über Partner
/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))], MESSAGE=KINFL[_F=Dateiname][_p] $D_KINFL[_F=Dateiname][_p][,DB.xxxxxx]	Information über Sperren
/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))], MESSAGE=KINFP[_p] $D_KINFP[_p][,DB.xxxxxx]	Info über Transaktionen in PTC (Prepare-To-Commit)
/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))], MESSAGE=KINFS[_p] $D_KINFS[_p][,DB.xxxxxx]	Informationen über Stati
/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))], MESSAGE=KINFT[_p] $D_KINFT[_p][,DB.xxxxxx]	Information über Trans-aktionen
/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))], MESSAGE=KNTASKS_n $D_KNTASKS_n[,DB.xxxxxx]	Anzahl Tasks vorgeben
/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))], MESSAGE=KPEND_p $D_KPEND_p[,DB.xxxxxx]	PTC beenden
/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))], MESSAGE=KPRESET_p $D_KPRESET_p[,DB.xxxxxx]	PTC rücksetzen

```
$D_KENA_x[,DB.xxxxxx]
```

Steuerung von ENA-COMM.

Wird ENA-COMM (enable communication) gesperrt, so kann sich kein neuer Partner an CISKOOR anschließen.

$D	Operation für "Diagnose"
KENA	Operationsergänzung für Steuerung von ENA-COMM.
x	- L (Lock) : ENA-COMM sperren - U (Unlock) : ENA-COMM wieder freigeben - D (Display) : Den aktuellen Zustand ausgeben
xxxxxx	Die Angabe eines Paßwortes DB.xxxxxx wirkt sich erst auf das nächste Kommando aus.

Quittung am Terminal: CISKOOR - O.K. wenn Kommando ausgeführt wurde.

CISKOOR - STATE: xxxx bei Anforderung der Auflistung mögliche Werte:

OK	: alles O.K.
OKLE	: O.K. aber ENA gesperrt
ENDE	: Sitzung-beendet
LADE	: CISKOOR wird geladen
RST	: Restart läuft
FRST	: Fehler bei Restart
KILL	: Kill-Kommando gegeben

```
$D_KEND_b[,DB.xxxxxx]
```

Setzen der Beendigungsbedingung.

$D Operation für "Diagnose".

KEND Operationsergänzung für Beendigungsbedingung.

b Beendigungsbedingung:

- 1 = CISKOOR wird nach Abmeldung des letzten Partners beendet.
- 2 = CISKOOR kann während der Betriebszeit nicht beendet werden. Außerhalb der Start- und Endezeit wird CISDBH auf die Beendigungsbedingung 1 gesetzt und nach Abmeldung des letzten Partners beendet.
- 3 = CISKOOR kann nicht beendet werden. Zur Beendigung muß mit einem nachfolgenden Kommando die Beendigungsbedingung auf 1 oder 2 gesetzt werden.

xxxxxx Die Angabe eines Paßwortes DB.xxxxxx wirkt sich erst auf das nächste Kommando aus.

Quittungen am Terminal:

- CISKOOR - O.K. wenn neue Beendigungsbedingung gesetzt ist.
- CISKOOR - ERROR wenn falsche Beendigungsbedingung angegeben wurde.

```
$D_KINFB[_p][,DB.xxxxxx]
```

Information über Before-Images.

$D	Operation für "Diagnose".
KINFB	Operationsergänzung für Information über Before-Images.
p	Position in der internen Transaktionstabelle, ab der protokolliert wird. Wird p nicht angegeben, wird immer ab Position 1 ausgegeben.
xxxxxx	Die Angabe eines Paßwortes DB.xxxxxx wirkt sich erst auf das nächste Kommando aus.

Beispiel einer Auflistung:

```
**** INFO-BIM . ****************************
*POS HOSTNAME APPLINAM *****TID MODE N.PAGES
   1 *                   4019       N        0
   2 *        CIS        DS01       U        1
END  (100)
```

Erläuterung der Datenzeilen:

POS	Position. Nach der letzten Datenzeile erscheint: END (xxx). xxx = max. Anzahl der angeschlossenen Partner. (siehe auch Parameter KNTRANS - Seite 224)
HOSTNAME	Hostname (* = eigener Prozessor).
APPLINAM	Name der Anwendung (Spaces bei TIAM).
TID	Transaktionskennung.
MODE	Betriebsart des Partners: N = Teilnehmer (TIAM) U = UTM (nicht synchronisiert) S = UTM-synchronisiert
PAGES	Anzahl beschriebener PAM-Pages in der Before-Image-Datei.

```
$D_KINFC[_p][,DB.xxxxxx]
```

Information über Partner.

$D	Operation für "Diagnose".
KINFC	Operationsergänzung für Information über Partner.
p	Position in der internen Transaktionstabelle, ab der protokolliert wird. Wird p nicht angegeben, wird immer ab Position 1 ausgegeben.
xxxxxx	Die Angabe eines Paßwortes DB.xxxxxx wirkt sich erst auf das nächste Kommando aus.

Beispiel einer Auflistung:

```
**** INFO-COMM. *******JJJJN 1 0700 1800*003/001***172243
*POS *TSN USER-ID. **MP-ADR SIZE(4K) MODE          **TIME
     5046 VD5DB    00020000
   1 5044 VD5DB    00021000 00000009 N 11          172111
END  (100)
```

Erläuterung der Überschriftzeile:

JJJJN	Status der folgenden CISKOOR-Parameter:	After-Image-Datei:	J
		Before-Image-Datei:	J
		Protokolldatei:	J
		Transaktionen:	J
		Diagnose-Ein:	N

1	CISKOOR-Beendigungsbedingung.
0700	Start der Betriebszeit.
1800	Ende der Betriebszeit.
003/001	max.Anzahl/akt.Anzahl CISKOOR-Tasks.
172243	Uhrzeit der Protokollierung (Stunde, Minuten, Sekunden).

Erläuterung der Datenzeilen:

POS	Position in der internen Tabelle der Partner. Die 1. Zeile betrifft immer CISKOOR selbst und hat keine Positionsnummer. Nach der letzten Datenzeile erscheint: END (xxx). xxx = max. Anzahl der angeschlossenen Partner. (vgl. auch Parameter KNCOMM).
TSN	Task-Sequence-Number des Partners (bzw. von CISKOOR in der ersten Zeile).
USER-ID	Benutzerkennung des Partners (bzw. von CISKOOR in der 1. Zeile).
MP-ADR	Adresse des Bereichs für diesen Partner im Common-Memory-Pool.
SIZE	Größe des Bereichs im Common-Memory-Pool in Vielfachem von 4K.
MODE	Betriebsart des Partners: N = Teilnehmer (TIAM) U = UTM (nicht synchronisiert) S = UTM-synchronisiert es folgt die CIS-Version des Partners.
TIME	Uhrzeit der Anmeldung des Partners (Stunde, Minuten, Sekunden)

```
$D_KINFL[_F=Dateiname][_p][,DB.xxxxxx]
```

Information über Sperren.

$D	Operation für "Diagnose".
KINFL	Operationsergänzung für Informationen über Sperren.
Dateiname	Name einer Datei, über die Sperrinformationen ausgegeben werden sollen. Wird keine Datei angegeben, so werden alle Informationen aufgelistet.
p	Position in der internen Transaktionstabelle, ab der protokolliert wird. Wird p nicht angegeben, so wird immer ab Position 1 ausgegeben.
xxxxxx	Die Angabe eines Paßwortes DB.xxxxxx wirkt sich erst auf das nächste Kommando aus.

Beispiel einer Auflistung:

```
**** INFO-LOCK-CISKOOR ******************** ... ***************
*POS HOSTNAME APPLINAM *****TID *FILE-NAME ... FF **REC. **FILE
   1 *                 5044     $UID.HD.   ...  A     4       1
END  (100)
```

Erläuterung der Datenzeilen:

POS	Position in der internen Tabelle der Transaktionen. Nach der letzten Datenzeile erscheint: END (xxx) xxx = max. Anzahl der aktuellen Transaktionen. (vgl. auch Parameter KNTRANS).		
HOSTNAME	Name des Rechners in dem die Transaktion ihren Ursprung hat. (* = eigener Rechner).		
APPLINAM	Name der Anwendung in der die Transaktion ihren Ursprung hat. Stehen hier Spaces, so ist der Partner ein Teilnehmerprogramm.		
TID	Transaktions-Identifikation:	bei Betriebsart N: bei Betriebsart U: bei Betriebsart S:	TSN plus 4 Spaces logischer Terminalname (KCLOGTER) Vorgang-Nummer (TAMVGNR) abdruckbar plus 4 Spaces
FILE-NAME	Dateiname (Name der HD)		
FF	Diverse Flags:	1. Byte: 2. Byte:	Space H : HD gesperrt R : reserviert V : Vor-reserviert A : Aufsteigend R : Random
REC	Anzahl gesperrter Sätze.		
FILE	Anzahl Sperrblöcke, die bei vollem Sperrspeicher in die Datei ausgelagert werden.		

```
$D_KINFP[_p][,DB.xxxxxx]
```

Information über Transaktionen in PTC.

$D	Operation für "Diagnose".
KINFP	Operationsergänzung für Information über Transaktionen im Zustand PTC.
p	Position in der internen Transaktionstabelle, ab der protokolliert wird. Wird p nicht angegeben, so wird immer ab Position 1 ausgegeben.
xxxxxx	Die Angabe eines Paßwortes DB.xxxxxx wirkt sich erst auf das nächste Kommando aus.

Beispiel einer Auflistung:

```
**** INFO-PETA-CISKOOR ***************************************
*POS FLAG HOSTNAME APPLINAM **************TID **ID
   1 NU   *                  4711             0001
===> INFO-TRANS-CISKOOR
END  (400)
```

Erläuterung der Datenzeilen:

POS	Position in der internen Tabelle der Transaktionen. Nach der letzten Datenzeile erscheint: END (xxx). xxx = Anzahl Einträge in der Tabelle. (siehe Parameter KNSTA).
FLAG	Diverse Kennungen (4 Bytes): 1. Byte : Betriebsart (S, U, N) 2. Byte : Herkunft (zur Zeit immer U) 3. Byte : leer 4. Byte : leer
HOSTNAME	Name des Prozessors.
APPLINAM	Name der Anwendung.
TID	Transaktionskennung
ID	Interne Kennung
===> INFO-TRANS-CISKOOR	Es sind Transaktionen, die in der Transaktionstabelle enthalten sind, im Zustand PTC vorhanden.

```
$D_KINFS[_p][,DB.xxxxxx]
```

Information über Stati.

$D	Operation für "Diagnose".
KINFS	Operationsergänzung für Information über Stati.
p	Position in der internen Transaktionstabelle, ab der protokolliert wird. Wird p nicht angegeben, so wird immer ab Position 1 ausgegeben.
xxxxxx	Die Angabe eines Paßwortes DB.xxxxxx wirkt sich erst auf das nächste Kommando aus.

Beispiel einer Auflistung:

```
**** INFO-STATI-CISKOOR ***************************************
*POS FLAG HOSTNAME APPLINAM PAM-PAGE SC   #TAS 1ST TID LAST TID
   1 NU *                  00000002 00      1 4711
END  (400)
```

Erläuterung der Datenzeilen:

POS	Position in der internen Tabelle der Transaktionen. Nach der letzten Datenzeile erscheint: END (xxx). xxx = Anzahl Einträge in der Tabelle. (siehe Parameter KNSTA).	
FLAG	Diverse Kennungen (4 Bytes):	1. Byte : Betriebsart (S, U, N) 2. Byte : Herkunft (zur Zeit immer U) 3. Byte : leer 4. Byte : leer
HOSTNAME	Name des Prozessors	
APPLINAM	Name der Anwendung	
PAM-PAGE	Pam-Page in der Status-Datei	
SC UTM	Session-Counter	
#TAS	Anzahl Transaktionen	1ST TID : Erste gespeicherte TID LAST TID : Letzte gespeicherte TID

```
$D_KINFT[_p][,DB.xxxxxx]
```

Information über Transaktionen.

$D	Operation für "Diagnose".
KINFT	Operationsergänzung für Information über Transaktionen.
p	Position in der internen Transaktionstabelle, ab der protokolliert wird. Wird p nicht angegeben, so wird immer ab Position 1 ausgegeben.
xxxxxx	Die Angabe eines Paßwortes DB.xxxxxx wirkt sich erst auf das nächste Kommando aus.

Beispiel einer Auflistung:

```
**** INFO-TRANS-CISKOOR ************************************
*POS HOSTNAME APPLINAM **TID ****UTM-TID SP. *TSN MODE *USER
   1 *                 5044                P  5044  N    50444
END  (100)
```

Erläuterung der Datenzeilen:

POS	Position in der internen Tabelle der Transaktionen. Nach der letzten Datenzeile erscheint: END (xxx). xxx = max. Anzahl der Transaktionen. (siehe auch Parameter KNTRANS).		
HOSTNAME	Name des Rechners in dem die Transaktion ihren Ursprung hat. (* = eigener Rechner).		
APPLINAM	Name der Anwendung in der die Transaktion ihren Ursprung hat. Stehen hier Spaces, so ist der Partner ein Teilnehmerprogramm.		
TID	Transaktions-Identifikation:	bei Betriebsart N: bei Betriebsart U: bei Betriebsart S:	TSN plus 4 Spaces logischer Terminalname (KCLOGTER) Vorgang-Nummer (TAMVGNR) abdruckbar plus 4 Spaces
UTM-TID	Volle UTM Transaktions-Identifikation:	bei Betriebsart N: Spaces bei Betriebsart U: Spaces bei Betriebsart S: nach folgender Tabelle:	

Bytes	Länge	Inhalt	Ursprung
0- 1	2	Session-Zähler	TAMSESCN
2- 7	6	Vorgangszähler	TAMVGCNT
8-11	4	Transaktionsnummer	TAMTACNT
12-15	4	Vorgangsnummer	TAMVGNR

Alle Felder hexadezimal abdruckbar.

SP. Spezial-Info: P: im Zustand PTC (= Prepare To Commit).
W: im Zustand PTC nach Warmstart.

TSN TSN des Partners, indem die Transaktion ihren Ursprung hat. Bei Betriebsart U und S stehen hier Spaces.

MODE Betriebsart: N = Teilnehmerbetrieb (TIAM)
U = UTM (nicht-synchronisiert)
S = UTM-synchronisiert

USER Name des UTM-Users oder des logischen Terminals. Bei Betriebsart N steht hier die TSN.

```
$D_KNTASKS_n[,DB.xxxxxx]
```

Setzen der Anzahl Tasks.

$D Operation für "Diagnose".

KNTASKS Operationsergänzung für Setzen Anzahl Tasks.

n Neue Anzahl der Tasks.

Ist n größer als der bisherige Wert, so werden weitere Tasks gestartet. Hierzu werden in CISKOOR /ENTER-JOB Kommandos abgesetzt. Der CISKOOR-Parameter KENT-JOB muß vorhanden sein.

Ist n kleiner als der bisherige Wert, so werden Tasks beendet, bis nur noch die neue Anzahl Tasks aktiv ist. Die Tasks werden immer nur nach dem Ende eines Zyklus beendet.

n kann nicht größer als der Maximalwert (KMAXT=) werden, und auch nicht kleiner als 1. Wird einer dieser Werte angegeben, so wird ohne Fehlermeldung max. bzw. 1 angenommen.

xxxxxx Die Angabe eines Paßwortes DB.xxxxxx wirkt sich erst auf das nächste Kommando aus.

Quittungen am Terminal: CISKOOR - O.K. nach Ausführung des Kommandos.
CISKOOR - ERROR nach Fehler bei /ENTER-JOB.

```
$D_KPEND_p[,DB.xxxxxx]
```

PTC beenden.

$D	Operation für "Diagnose".
KPEND	Operationsergänzung für PTC beenden Die angegebene Transaktion (siehe p) wird beendet.
p	Position der Transaktion im Ausdruck beim Aufruf KINFP (vgl. Ausdruck Seite 235).
xxxxxx	Die Angabe eines Paßwortes DB.xxxxxx wirkt sich erst auf das nächste Kommando aus.

Quittungen am Terminal: CISKOOR - O.K. wenn Transaktion beendet.
CISKOOR - ERROR nach Fehler.

```
$D_KPRESET_p[,DB.xxxxxx]
```

PTC rücksetzen.

$D	Operation für "Diagnose".
KPRESET	Operationsergänzung für PTC rücksetzen. Die angegebene Transaktion (siehe p) wird zurückgesetzt.
p	Position der Transaktion beim Aufruf KINFP.
xxxxxx	Die Angabe eines Paßwortes DB.xxxxxx wirkt sich erst auf das nächste Kommando aus.

Quittungen am Terminal: CISKOOR - O.K. wenn Transaktion rückgesetzt.
CISKOOR - ERROR nach Fehler.

6 CISINIT

6.1 Funktionsumfang

CISINIT initialisiert die Dateien der Datensicherung:

- After-Image-Datei
- Before-Image-Datei
- Protokolldatei

Bei der Initialisierung wird der Inhalt der Dateien auf einen Anfangswert gesetzt. Dies ist notwendig bevor die Dateien das erste Mal benützt werden.

Die After-Image-Datei und die Protokolldatei müssen mit CISINIT zu bestimmten Zeitpunkten rückgesetzt werden. Die Initialisierung ist notwendig, da CISKOOR diese Dateien immer fortschreibt. (Zur After-Image-Datei vgl. auch Abschnitt über CISRA - Seite 245). Sinnvollerweise sollte die Initialisierung nach der Komplettsicherung durch das Rechenzentrum (meist wöchentlich) durchgeführt werden.

Die Before-Image-Datei darf nicht mit CISINIT bearbeitet werden, es sei denn, ein Kaltstart ist notwendig. Wird CISKOOR korrekt beendet, so ist die Before-Image-Datei leer. Nach einem Crash jedoch können Sätze in der BIM-Datei vorhanden sein. Diese Sätze müssen von CISKOOR verarbeitet werden, um die Dateien wieder in einen konsistenten Zustand zu versetzen.

6.2 Bedienung

CISINIT wird über SYSDTA gesteuert. Die Meldungen werden über SYSOUT ausgegeben.

6.3 Steueranweisungen

Als Antwort auf die Meldung `PARAMS (AIM/BIM/PRO/END)?` können folgende Anweisungen eingegeben werden:

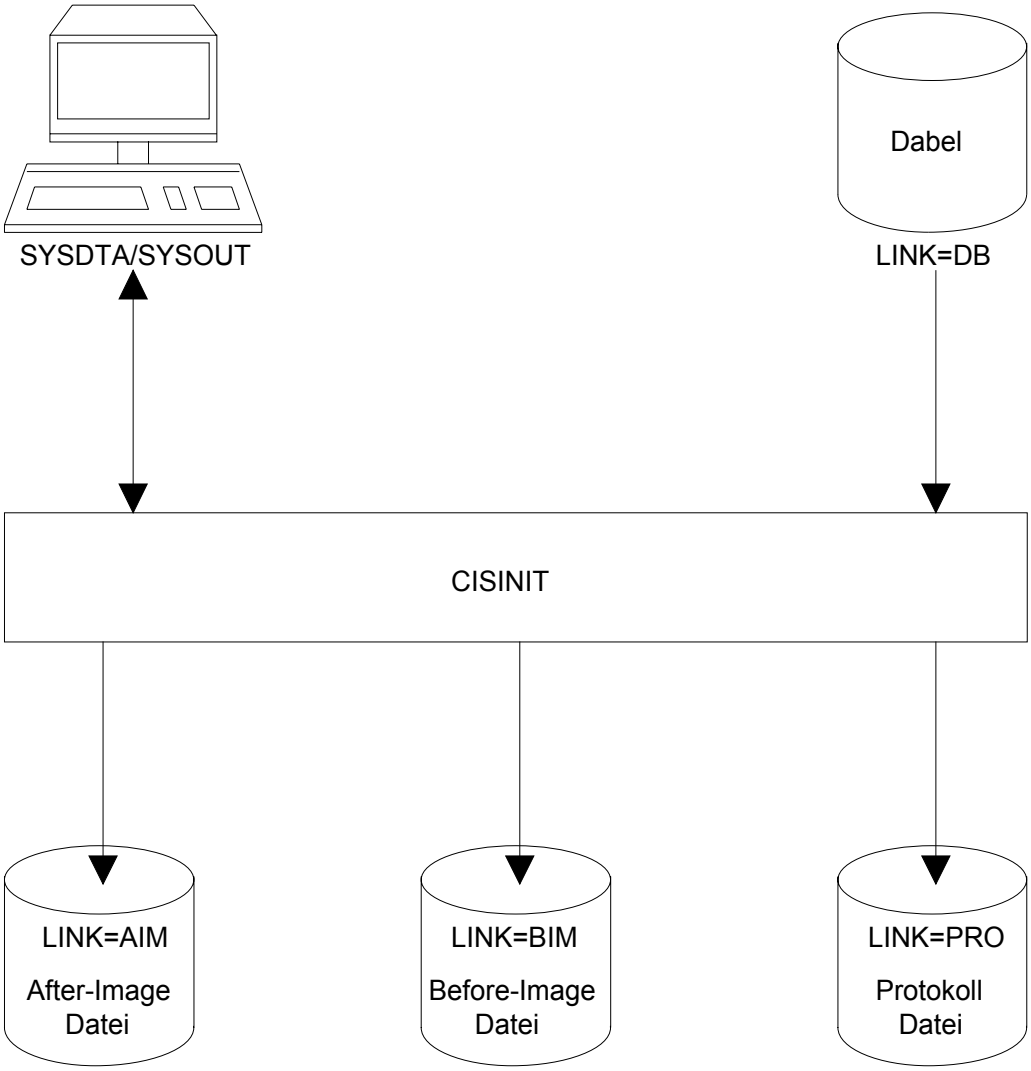
AIM Initialisierung der After-Image-Datei.

BIM Initialisierung der Before-Image-Datei. (Ab CIS V11 entfällt der Parameter BIM=nnn.)

PRO Initialisierung der Protokolldatei.

END Beendigung des Programms.

6.4 Datenflußdiagramm



6.5 Dateinamen

CISINIT benützt folgende Dateien:

Datei	LINK-Name	FCB-Typ	Standard-Dateiname
Dabel	DB	ISAM	DABEL
After-Image-Datei	AIM	PAM	CIS.AIM
Before-Image-Datei	BIM	PAM	CIS.BIM
Protokolldatei	PRO	ISAM	CIS.PRO

Wird als Dateiname der vorgegebene Dateiname verwendet, so muß die jeweilige Datei nicht mit dem LINK-Parameter zugewiesen werden.

6.6 Beispiel für den Ablauf von CISINIT

Dialog am Terminal:

```
/CREATE-FILE FILE-NAME=CIS.AIM,SUPPORT=PUBLIC-DISK(SPACE=RELATIVE
(PRIMARY-ALLOCATION=300,SECONDARY-ALLOCATION=30))
/CREATE-FILE FILE-NAME=CIS.BIM,SUPPORT=PUBLIC-DISK(SPACE=RELATIVE
(PRIMARY-ALLOCATION=300,SECONDARY-ALLOCATION=30))
/CREATE-FILE FILE-NAME=PROTOKOLL,SUPPORT=PUBLIC-DISK(SPACE=RELATIVE
(PRIMARY-ALLOCATION=300,SECONDARY-ALLOCATION=30))
/SET-FILE-LINK FILE-NAME=PROTOKOLL,LINK-NAME=PRO
/START-PROGRAM FROM-FILE=CISINIT
% P500 LOADING
CI50   CISINIT Vxx.x VOM xx.xx.xx
CI52   PARAMS (AIM/BIM/PRO/END) ?
*
AIM
CI52   PARAMS (AIM/BIM/PRO/END) ?
*
BIM
CI52   PARAMS (AIM/BIM/PRO/END) ?
*
PRO
CI52   PARAMS (AIM/BIM/PRO/END) ?
*
END
CI51   ENDE CISINIT
/
```

Sollen diese drei Dateien neu initialisiert werden, so kann folgende Prozedur benützt werden:

```
/BEGIN-PROCEDURE
/SET-FILE-LINK FILE-NAME=DABEL.USER,LINK-NAME=DB
/SET-FILE-LINK FILE-NAME=PROTOKOLL,LINK-NAME=PRO
/ASSIGN-SYSDTA TO-FILE=*SYSCMD
/START-PROGRAM FROM-FILE=CISINIT
AIM
BIM
PRO
END
/ASSIGN-SYSDTA TO-FILE=*PRIMARY
/END-PROC
```

Im Beispiel entsprechen die Namen der After-Image-Datei und der Before-Image-Datei den standardmäßigen (von CIS vergebenen) Dateinamen, deshalb ist keine LINK-Zuweisung nötig. Für die Protokolldatei wird kein vorgegebener Dateiname verwendet, daher muß sie mit dem LINK-Namen zugewiesen werden.

Bemerkung:

Mit Hilfe der Prozedur D.CIS.DASI.INSTALLIEREN wird nach Wunsch auch eine Prozedur für das Initialisieren der Datensicherungsdateien im Dialog aufgebaut (vgl. Manual-1: Installieren der Datensicherung).

7 CISRA

7.1 Funktionsumfang

CISRA führt den Recovery mit den After-Image-Sätzen aus. Bevor CISRA abläuft, müssen die CIS-Hauptdateien und die CIS-Verweisdateien auf den Inhalt zurückgesetzt werden, den sie hatten, als die After-Image-Datei noch leer war.

7.2 Allgemeiner Ablauf

Beschreibung des allgemeinen Ablaufs der Datensicherung, um die Einordnung von CISRA besser verstehen zu können.

1. CIS-Dateien sichern: Regelmäßige Sicherung, z.B. wöchentliche Datensicherung.

2. After-Image-Datei initialisieren: Benützung des Programms CISINIT.

3. Änderungen in den Dateien.

   Im CIS-Betrieb werden die Hauptdateien und Verweisdateien geändert. Jede Änderung wird von CISKOOR in die After-Image-Datei geschrieben. Die Änderungen in den Verweisdateien können in die After-Image-Datei geschrieben werden. (Performance- und Sicherheitsüberlegungen).

4. Anforderung eines Vorsetzens.

   Ein Vorfall hat die Daten einer oder mehrerer Dateien zerstört. Der Betrieb muß angehalten werden, die Daten müssen wieder hergestellt werden.

5. Sicherungsdaten lesen.

   Die zerstörten Daten werden mit Hilfe der Sicherungsdateien repariert. Der Stand der Daten ist jetzt der, den sie zum Zeitpunkt des Sicherungslaufs hatten. Ab diesem Zeitpunkt wurden After-Image-Sätze gespeichert.

   Daraus ergibt sich:

6. CISRA-Lauf.

   Das Programm CISRA liest die Änderungen in der After-Image-Datei und schreibt sie in die Hauptdateien und eventuell Verweisdateien. Hier wird natürlich angenommen, daß die After-Image-Datei nicht zerstört ist. Es können alle Dateien bearbeitet werden für die After-Image-Sätze vorhanden sind.

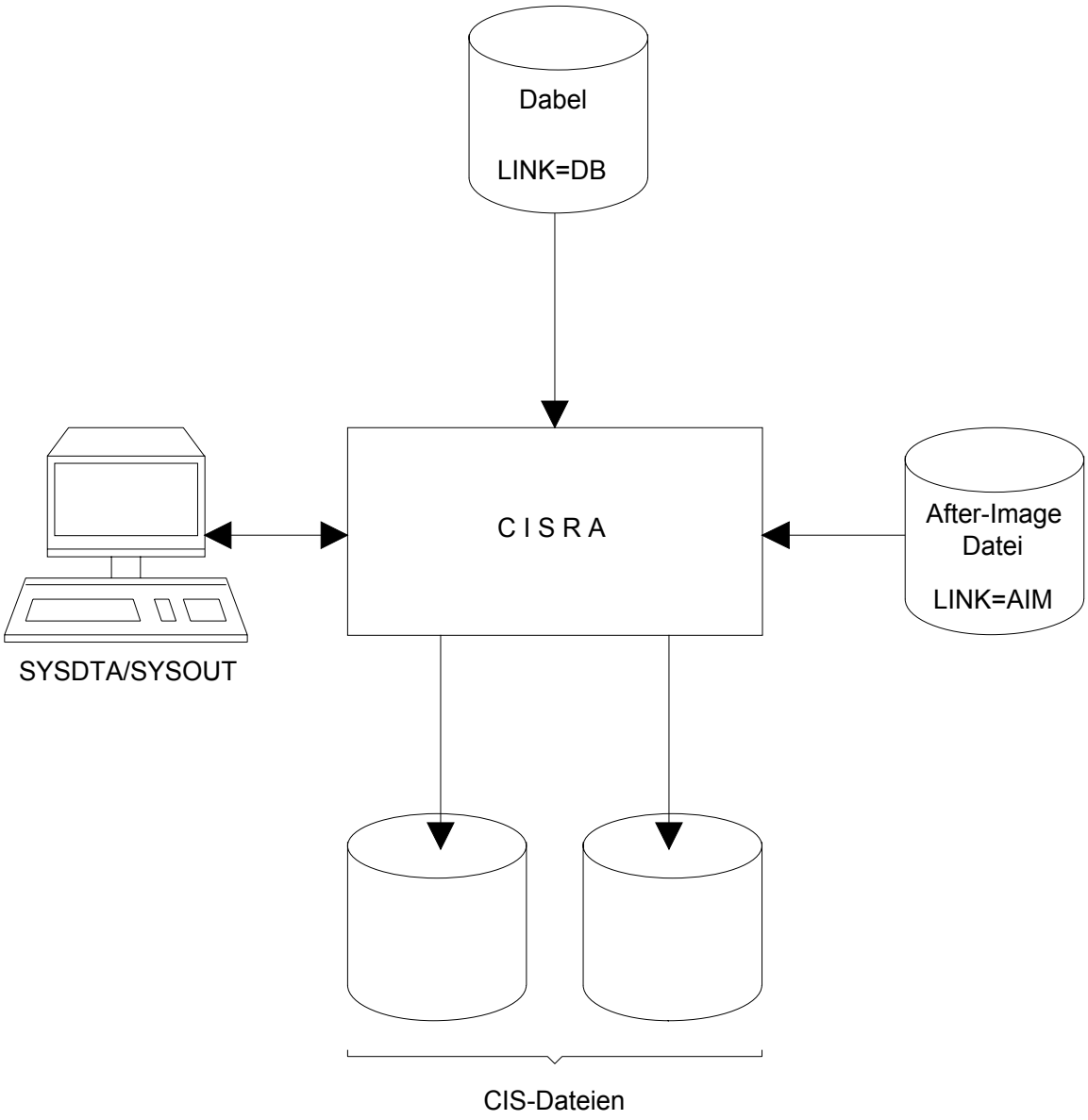
7. Verweisdateien bearbeiten.

   Die Verweisdateien müssen, wenn keine After-Image-Sätze für sie vorhanden sind, mit dem Programm CISLADF neu erstellt werden.

8. Zustand der Daten.

   Jetzt sind die Daten auf dem Stand, den sie zum Zeitpunkt des Ausfalls hatten. Die Änderungen von noch offenen Transaktionen sind zurückgesetzt.

7.3 Datenflußplan von CISRA



7.4 Bedienung

CISRA wird über SYSDTA gesteuert. Die Meldungen werden über SYSOUT ausgegeben.

7.4.1 Steueranweisungen

```
PARAMS ? (TID=/DAT=/END)
```

TID=n Anzahl (n max. 6-stellig) verschiedener TID's in der After-Image-Datei.
Vorgegebener Wert: 2000; kleinere Werte werden ignoriert.

TID = Transaktions-ID im Teilnehmerbetrieb: TSN + 4 Spaces
im Teilhaberbetrieb: logische Terminalnummer
UTM-interne Nummer

DAT=n Anzahl (n max. 6-stellig) verschiedener CIS-Dateien, für die After-Images in der After-Image-Datei enthalten sein können. Vorgegebener Wert: 1000; kleinere Werte werden ignoriert.

END Ende der Parametereingabe.

Bemerkung:

Die Parameter 'TID' und 'DAT' bestimmen die Größe von internen Tabellen. Die Größe der Tabelle errechnet sich folgendermaßen:

t * 48 + d * 572 t = Anzahl TID's (TID=)
d = Anzahl Dateien (DAT=)

```
DATEISELEKTION ? (SEL/ALL/LIST)
```

ALL Alle Dateien, für die After-Images vorhanden sind, werden rekonstruiert.

SEL Auswahl der Dateien die zu rekonstruieren sind. Es erscheint immer wieder die Meldung:

```
NAME-O(,NAME-K) ODER END
```

Diese Meldung ist zu beantworten mit: 'Name der Originaldatei[,Name der Kopie]' oder 'END'.

Die selektierten Dateien, für die After-Images vorhanden sind, werden rekonstruiert.

Sind zwei Dateinamen angegeben, so werden die After-Images, die für die erste Datei (Original) erzeugt wurden in die zweite Datei (Kopie) geschrieben.

Alle Dateinamen sind vollqualifiziert (mit 'catid' und 'user-id') einzugeben. Sind die Namen nicht vollständig bekannt, so kann die Funktion LIST aufgerufen werden.

LIST Es soll eine Liste aller vorhandener Dateien ausgegeben werden. Die in der Liste ausgegebenen Namen müssen bei der Dateiselektion in der gleichen Art eingegeben werden, also mit 'catid' und 'user-id'.

7.4.2 Meldungen

ANFANG PRUEFUNG — CISRA geht die After-Image-Datei durch und prüft, ob die einzelnen Transaktionen komplett sind. Nicht komplette Transaktionen werden nicht verarbeitet.

ANFANG UPDATE — Die After-Image-Sätze werden zurückgeschrieben.

```
ENDE UPDATE
SAETZE GEAENDERT Dateiname
```

Ende des CISRA-Laufs: Die bearbeiteten CIS-Dateien (DATEINAME) werden aufgelistet. Für jede Datei wird die Zahl der gesamten After-Image-Sätze (SAETZE) und die Zahl der korrekt verarbeiteten After-Image-Sätze (GEAENDERT) ausgegeben.

Am Ende wird eine Summenzeile ausgegeben.

Vgl. auch Seite 250 und 251.

7.4.3 Dateinamen

CISRA benötigt folgende Dateien:

Datei	LINK-Name	FCB-Typ	Standard-Dateiname
Dabel	DB	ISAM	DABEL
After-Image-Datei	AIM	PAM	CIS.AIM
CIS-Dateien		ISAM	Name aus AIM-Satz

Wird als Dateiname der standardmäßige Dateiname verwendet, so muß die jeweilige Datei nicht mit dem LINK-Parameter zugeweisen werden.

Beispiele:

CISRA-Lauf ohne Selektion:

```
/SET-FILE-LINK FILE-NAME=USERDABEL,LINK-NAME=DB
/START-PROGRAM FROM-FILE=CISRA
%  P500 LOADING
RA50    CISRA Vxx.x VOM xx.xx.xx
RA59    PARAMS ? (TID=/DAT=/END)
*
END
RA60    DATEISELEKTION ? (SEL/ALL/LIST)
*
ALL
RA53    ANFANG PRUEFUNG
RA54    ANFANG UPDATE
RA55    ENDE UPDATE
RA56    SAETZE GEAENDERT  DATEINAME
            1         1  $CIS11.HD.MESSE1
            2         2  $CIS11.VD.MESSE1
            3         3  -----------------

RA58    PROGRAMM-ENDE
/
```

CISRA-Lauf mit Selektion:

```
/START-PROGRAM FROM-FILE=CISRA
%  P500 LOADING
RA50    CISRA Vxx.x VOM xx.xx.xx
RA59    PARAMS ? (TID=/DAT=/END)
*
END
RA60    DATEISELEKTION ? (SEL/ALL/LIST)
*
SEL
RA61    NAME-O(,NAME-K) ODER END
*
:O:$CIS11.HD.MESSE1,:O:$CIS11.HD.MESSE1.COPY
RA61    NAME-O(,NAME-K) ODER END
*
END
RA53    ANFANG PRUEFUNG
RA54    ANFANG UPDATE
RA55    ENDE UPDATE
RA56    SAETZE GEAENDERT  DATEINAME
           10        10  :O:$CIS11.HD.MESSE1.COPY
           10        10  ----------------------

RA58    PROGRAMM-ENDE
/
```

Erläuterung: Während eines oder mehrerer CIS-Läufe wurde die Datei verändert. Für die Datei gibt es After-Image-Sätze. Mit CISRA werden die Änderungen in die Datei HD.MESSE1.COPY geschrieben.

CISRA-Lauf mit Auflistung:

```
/START-PROGRAM FROM-FILE=CISRA
%  P500 LOADING
RA50    CISRA Vxx.x VOM xx.xx.xx
RA59    PARAMS ? (TID=/DAT=/END)
*
END
RA60    DATEISELEKTION ? (SEL/ALL/LIST)
*
LIST
RA53    ANFANG PRUEFUNG
RA56    SAETZE GEAENDERT  DATEINAME
            1         0  :O:$CIS11.HD.NITA01
            1         0  :O:$CIS11.HD.MESSE1
            1         0  :O:$CIS11.HD.FMULTI
            3         0  ----------------------

RA58    PROGRAMM-ENDE
/
```

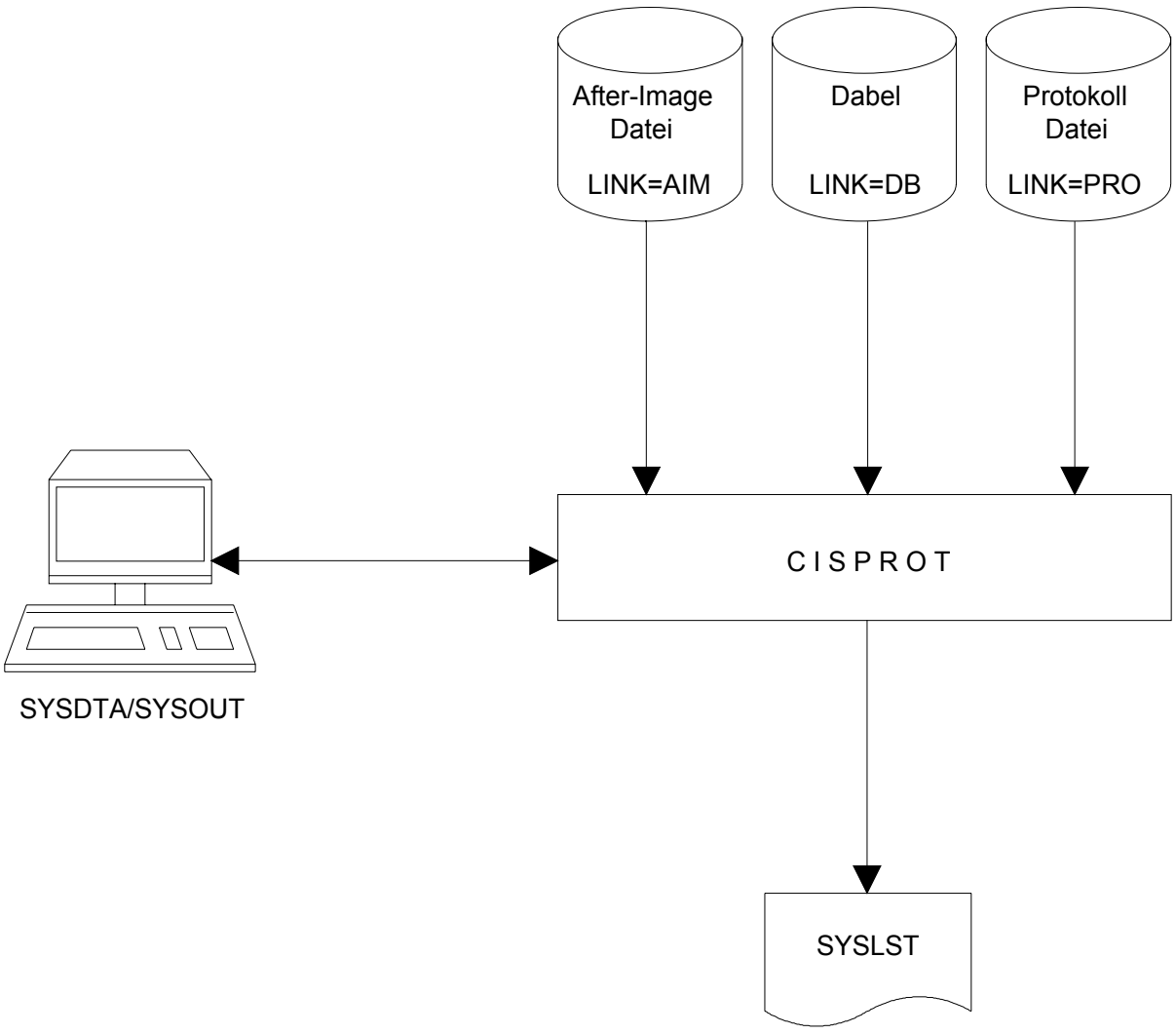
Erläuterung: Dieser Lauf wird z.B. ausgeführt wenn nicht genau bekannt ist für welche Dateien es After-Image-Sätze gibt, oder wenn z.B. 'catid' oder 'user-id' nicht genau bekannt sind.

8 CISPROT

8.1 Funktionsumfang

CISPROT druckt die After-Image-Datei und die Protokolldatei aus. Die auszudruckenden Sätze können nach bestimmten Kriterien selektiert werden.

8.2 Datenflußplan von CISPROT



8.3 Bedienung

CISPROT wird über SYSDTA gesteuert. Die Meldungen werden über SYSOUT ausgegeben. Die Liste wird über SYSLST ausgedruckt.

8.3.1 Steueranweisungen

Als Antwort auf die Meldung PARAMS (AIM/PRO/END)? können folgende Anweisungen eingegeben werden:

AIM	After-Image-Datei ausdrucken.
PRO	Protokolldatei ausdrucken.
END	Programm beenden.

Bei Druck der After-Image-Datei werden folgende Parameter angefordert:

ALLES/FEHLER (ALL/ERR)?	ALL	Alle Sätze ausgeben.
	ERR	Es werden nur die Sätze ausgeben, die nicht von CISRA verarbeitet werden können.
KOPF/DATEN (KOPF/DATEN)?	KOPF	Nur die Informationen aus dem Kopf des After-Image-Satzes drucken.
	DATEN	Kopf und Daten ausdrucken.

8.3.2 Selektionen

`SELEKTION (J/N)?`

- `J` Sätze selektieren.
- `N` Alle Sätze ausdrucken.

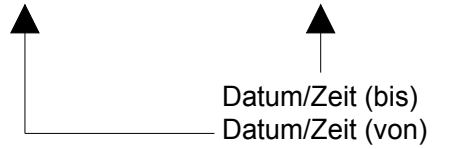
`DATUM/ZEIT-SELEKTION?`

Poolinit

- `J` Sätze nach Datum und Uhrzeit selektieren.
- `N` Alle Sätze für jedes Datum und jede Uhrzeit ausdrucken.

`DATUM/ZEIT (VON) - DATUM/ZEIT (BIS)?`

```
jjmmtt/hhmmss - jjmmtt/hhmmss
```



Datum und Uhrzeit ist der Zeitpunkt des Starts von CISKOOR.

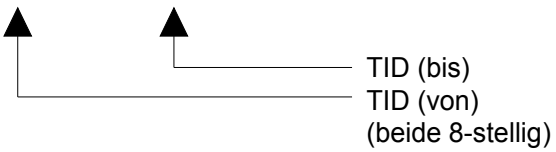
Format:			
	`jj`	Jahr	(92-99; 00-...)
	`mm`	Monat	(01-12)
	`tt`	Tag	(01-31)
	`hh`	Stunde	
	`mm`	Minuten	
	`ss`	Sekunden	

`TID-SELEKTION (J/N)?`

- `J` Sätze nach TID selektieren.
- `N` Sätze für jede TID ausdrucken.
- `TID` TRANSAKTIONS-ID:
  Teilnehmerbetrieb: TSN + 4 Spaces
  Teilhaberbetrieb: logische Terminalnummer

`TID (VON) - TID (BIS)?`

```
tttttttt-tttttttt
```



TID: Im Satz gespeicherte TRANSAKTIONS-ID.

`TSN-SELEKTION (J/N)?`

`J` Sätze nach TSN selektieren.

`N` Sätze für jede TSN ausdrucken.

`TSN (VON) - TSN (BIS)?`

`tttt-tttt` (beide 4-stellig)

TSN ist die im Satz gespeicherte TSN:

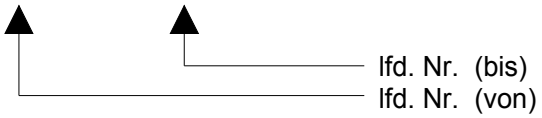
`LFD.NR.SELEKTION (J/N)?`

`J` Sätze nach laufender Nummer selektieren.

`N` Sätze für jede laufende Nummer ausdrucken.

`LFD.NR.(VON) - LFD.NR.(BIS)`

`llllllll-llllllll` (beide 8-stellig)



Das Format der laufenden Nummer ist immer 8-stellig sedezimal, z.B.: 000000A1-000001CD.

8.3.3 Dateinamen

CISPROT benützt folgende Dateien:

Datei	LINK-Name	FCB-Typ	Standarddateiname
Dabel	DB	ISAM	DABEL
After-Image-Datei	AIM	PAM	CIS.AIM
Protokolldatei	PRO	ISAM	CIS.PRO

Wird ein anderer Dateiname verwendet, so muß er mit dem entsprechenden LINK-Namen zugewiesen werden.

Beispiel:

Dialog am Terminal:

```
/SET-FILE-LINK FILE-NAME=CIS.PRO.T,LINK-NAME=PRO
/START-PROGRAM FROM-FILE=CISPROT
%  P500 LOADING
CP50    CISPROT Vx VOM xx.xx.xx
CP52    PARAMS (AIM/PRO/END)?
*
AIM
CP55    SELEKTION (J/N)?
*
N
CP70    ALLES/FEHLER (ALL/ERR)?
*
ALL
CP71    KOPF/DATEN (KOPF/DATEN)?
*
DATEN
```

(Druck der Liste)

```
CP52    PARAMS (AIM/PRO/END)?
*
PRO
CP55    SELEKTION (J/N)?
*
N
```

(Druck der Liste)

```
CP52    PARAMS (AIM/PRO/END)?
*
END
CP51    ENDE CISPROT
/
```

Beispiel für eine Prozedurdatei:

```
/BEGIN PROC
/SET-FILE-LINK FILE-NAME=CIS.PRO.T,LINK-NAME=PRO
/ASSIGN-SYSDTA TO-FILE=*SYSCMD
/START-PROGRAM FROM-FILE=CISPROT
AIM
N
ALL
DATEN
PRO
N
END
/ASSIGN-SYSDTA TO-FILE=*PRIMARY
/END-PROC
```

Im Beispiel entspricht der Name der After-Image-Datei dem Standardnamen, deshalb ist keine LINK-Zuweisung nötig. Der Name der Protokolldatei weicht vom Standardnamen ab, deshalb muß sie mit dem LINK-Namen zugewiesen werden.

9 CISSERV

9.1 Funktionsumfang

In diesem Kapitel wird der Aufbau der Client-Server-Architektur mit dem CIS-Server erläutert. Der Server ist ein BS2000-DCAM-Programm, das auch im Multitask-Betrieb laufen kann (multiple server). Die Clients sind Terminals oder Programme, die im Netz verteilt sein können. Sie werden vom Server im CIS-Aktiv-Modus bedient. Die ausgetauschten Daten sind also alle abdruckbar.

Die Clients von CISSERV können sein:

- BS 2000 Terminals
- CISDC auf einem SINIX-Rechner (MX-i oder RM)
- CISDC unter Windows

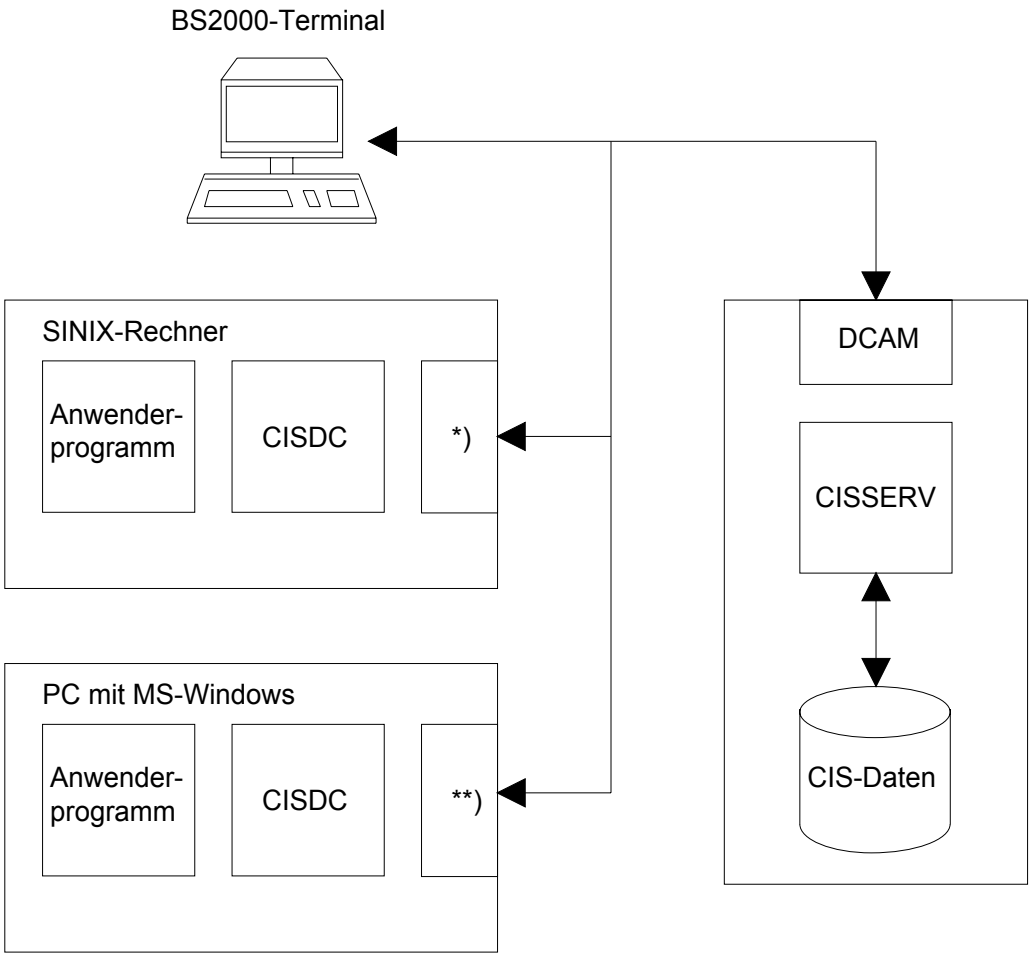
CISSERV ist eine DCAM-NEA Anwendung. Terminals können also ohne weiteres mit dem Server verkehren.

Im SINIX braucht das Modul CISDC das CMX mit dem NEABX-Migrationsservice.

Unter MS-Windows wird das NEA-Protokoll nachgebildet.

Wird CISSERV im Multitask-Betrieb gefahren, so kann der Primary-Task so eingestellt werden, daß er sich nur um die Kommunikation kümmert, selbst aber keine CIS-Routinen aufruft. Dies ist ein Sicherheitsaspekt, da ein DCAM-Primary, der sich fehlerhaft beendet, die ganze Anwendung beendet.

9.2 Datenfluß



*) CMX plus CCP

**) Emu + Driver + Hardware
oder
CMX + CCP + Hardware

9.3 Bedienung

CISSERV wird in einem ENTER-Prozeß gestartet.

Alle Anpassungen an die jeweiligen Belange werden über Parameter in diesem Prozeß gesteuert.

Eingriffe während des Betriebs können über das BS2000-Kommando '/SEND-MESSAGE' oder über das CIS-Kommando $D vorgenommen werden.

9.3.1 Einrichten von CISSERV

Es werden zwei Phasen von CISSERV ausgeliefert: CISSERV mit Datensicherung
CISSERV.ODASI ohne Datensicherung

Die dazugehörenden Bindeprozeduren werden mitgeliefert. Anpassungen an CISVARI werden ausgeführt wie bei CIS oder CIS.ODASI

- Anpassungen an CISVARI / CISVARI1
- Neu binden von CISSERV / CISSERV.ODASI

9.3.2 Start des CISSERV-Prozesses

Folgende ENTER-Prozedur muß definiert werden:

```
/LOGON
/SET-FILE-LINK FILE-NAME=Name,LINK-NAME=DB
/SET-FILE-LINK FILE-NAME=Name,LINK-NAME=KUKO
/START-PROGRAM FROM-FILE=CISSERV[.ODASI]
 .
 .                              (Parameter für CISSERV)
 .
END
/LOGOFF
```

9.3.3 Parameter für CISSERV

Übersicht:

Parameter	Bedeutung	Default-Wert
*...	Kommentar	
APPLI=a	Definition Anwendungsname	CIS
AUTEND	Automatisch beenden	
END	Ende Parameter	
ENT-JOB=params	Parameter zum Starten weiterer CISSERV-Tasks	
OUT-P=x	Ausgabemodus für Programme (F/L)	L
OUT-T=x	Ausgabemodus für Terminals (F/L)	L
POS=n	Anzahl Einträge in Partnertabelle	200
PRIMWAIT	Primary soll warten	
TASKS=n	Anzahl CISSERV-Tasks	1
PW-P=p	Paßwort für Programme	
PW-T=p	Paßwort für Terminal	
TIME-END=hhmn	Ende-Zeit	
TIME-STOP=hhmn	Abbruch-Zeit	
VERS-P=x	Versionsmeldung bei Programmen (J/N)	N
VERS-T=x	Versionsmeldung bei Terminals (J/N)	J

```
* ...
```

Kommentar - ohne Auswirkung.

Es können beliebig viele Kommentare vorhanden sein.

```
APPLI=a
```

Definition des Anwendungsprogramms von CISSERV.

Es kann ein 1- bis 8-stelliger Name angegeben werden. Fehlt der Parameter, so wird "CIS" angenommen.

```
AUTEND
```

Automatisches Ende.

Wenn der letzte Partner sich abmeldet, wird auch CISSERV beendet.

Fehlt der Parameter, so muß CISSERV mit dem TIME-Parameter oder durch Eingriff beendet werden.

```
END
```

Ende der Parameter.

```
ENT-JOB=params
```

Parameter zum Starten weiterer CISSERV-Tasks.

CISSERV schreibt die Parameter `params` in ein ENT-JOB Kommando (max. 200 Bytes lang und ruft dieses Kommando so oft auf bis der Wert von TASKS=N erreicht ist.

Fehlt der Parameter, so müssen zusätzliche Tasks manuell gestartet werden.

Intern wird das Kommando: `/ENTER-JOB FROM-FILE =params` aufgerufen.

```
OUT-P={F/L}
```

Ausgabemodus für Programme als Partner.

- F formatiert
- L Zeilenmodus (Line) = Standardwert

```
OUT-T={F/L}
```

Ausgabemodus für Programme auf Terminal.

- F formatiert
- L Zeilenmodus (Line) = Standardwert

```
POS=n
```

Anzahl Einträge in Partnertabelle.

Dies ist die Anzahl der Partner (Terminal oder CISDC-Verbindungen), die gleichzeitig mit CISSERV arbeiten können.

Fehlt dieser Parameter so wird der Wert 200 angenommen.

```
PRIMWAIT
```

Primary soll warten.

Wird dieser Parameter angegeben, so bearbeitet der Primary-Task keine CIS-Aufrufe. Er bedient lediglich die Kommunikation.

```
PW-P=p
```

Paßwort für Programme.

Paßwort, das bei Verbindungsaufbau von einem Partner, der ein Programm ist, angegeben werden muß. (Siehe auch Bearbeitung von CISDC)

Das Paßwort ist 1 bis 16 Zeichen lang.

```
PW-T=p
```

Paßwort für ein Terminal.

Das Anmelden des Terminals hat folgendes Format: `O_Name,...,MSG=c'PW=p'`

Das Paßwort ist 1 bis 16 Zeichen lang.

```
TASKS=n
```

Anzahl CISSERV-Tasks.

TASKS steuert das Laden zusätzlicher CISSERV-Tasks in Zusammenhang mit dem Parameter ENT-JOB=params.

Fehlt der Parameter, so wird der Wert 1 angenommen.

```
TIME-END=hhmn
```

Ende-Zeit

Ist diese Zeit erreicht, so geht CISSERV in den "AUTEND"-Zustand. (Siehe Parameter AUTEND.)

```
TIME-STOP=hhmn
```

Abbruch-Zeit

Ist diese Zeit erreicht, so wird CISSERV sofort beendet.

```
VERS-P={J/N}
```

Ausgabe der Versionsmeldung beim Anmelden für Programme als Partner.

J Ja
N Nein (Standardwert)

```
VERS-T={J/N}
```

Ausgabe der Versionsmeldung beim Anmelden des Terminals.

J Ja (Standardwert)
N Nein

9.3.4 Bedienung während des Betriebs

Die Bedienung erfolgt entweder über das BS2000 Kommando '/SEND-MESSAGE'oder über das CIS-Kommando $D.

Übersicht:

Kommando	Erläuterung
`/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))], MESSAGE=LST[_p]` `$DSLST[_p]`	Information über Partner
`/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))], MESSAGE=END` `$D_SEND`	Ende
`/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))], MESSAGE=STOP`	Abbruch

```
/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))],MESSAGE=LST[_p]
$D_SLST[_p]
```

Information über Partner.

Dieses Kommando bewirkt, daß eine Liste der aktiven Partner ausgegeben wird. Es werden jeweils 20 Zeilen gedruckt. Sollen weitere Partner aufgelistet werden, so wird beim Kommando die Position (p) angegeben, ab der protokolliert werden soll.

```
/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))],MESSAGE=END
$D_SEND
```

Ende

Mit diesem Kommando wird CISSERV in den Modus geschaltet, in dem es sich beendet wenn der letzte Partner sich abmeldet.

```
/SEND-MESSAGE TO=PROGRAM[(JOB-ID=TSN(xxxx))],MESSAGE=STOP
```

Abbruch

Mit diesem Kommando wird CISSERV sofort beendet.

9.3.5 Beendigung von CISSERV

Es gibt verschiedene Möglichkeiten, CISSERV zu beenden:

a) Parameter AUTEND oder /SEND-MESSAGE...END oder $D_SEND

CISSERV beendet sich wenn der letzte Partner sich abmeldet.

b) TIME-END=hhmn

Bei Erreichen dieser Zeit geht CISSERV in den Zustand wie unter a) beschrieben.

c) /SEND-MESSAGE...STOP oder /BCLOSE_Name oder /BCEND

CISSERV wird sofort beendet.

d) TIME-STOP=hhmn

Bei Erreichen dieser Zeit wird CISSERV sofort beendet.

10 CISDICT

10.1 Funktionsumfang

Das Dienstprogramm CISDICT ermöglicht Auskünfte nach verschiedenen Kriterien aus einer Datenbeschreibungsdatei in ähnlicher Form wie sie aus einem Data-Dictionary möglich sind. Die DABEL muß mit dem SET-FILE-LINK Kommando, LINK-NAME=DB, zugeordnet werden.

Folgende Funktionen stehen für die Ausgabe zur Verfügung:

- Allgemeine Übersichten
- Gezielte Auswertung über den HD-Namen
- Gezielte Auswertung über den Satzbeschreibungsnamen
- Gezielte Auswertung über den Transformations- /Bildbeschreibungsnamen
- Gezielte Auswertung über den Maskennamen
- Gezielte Auswertung über den Feldnamen
- Übersicht im Dateiformat (für Bildschirm/Liste bzw. Datei)

Die Ausgabe ist wahlweise auf Bildschirm (SYSOUT) oder auf Drucker (SYSLST) möglich. Die Form der Ausgabe ist die sogenannte Pseudografik. Bei der letzen Funktion kann die Ausgabe auch in eine Datei (Zuordnung über LINK-Namen) erfolgen, wobei die Ausgabeform zeilenorientiert ist, da eine Ausgabe in eine DVS-Datei im pseudografischen Format nicht sinnvoll ist.

Welche Teilfunktion für die Ausgabe durchgeführt werden soll, sowie das Ziel der Ausgabe werden in einer Anforderungsmaske eingegeben.

Unterschiedliche Schreibweisen des HD-Namens (mit/ohne CAT-/User-ID) werden logisch zusammengeführt.

Die Ausgabe auf Bildschirm und SYSLST ist im Prinzip identisch. Für SYSLST wird allenfalls ein vorzeitiger Seitenwechsel ausgelöst, um die Zerteilung einer Information zu verhindern.

Die Bildschirmausgabe erfolgt zeilenweise mittels WROUT-Makro, die Ausgabe auf SYSLST ebenfalls zeilenweise mittels WRLST-Makro. Die Anforderungsmaske wird mittels WRTRD-Makro im "extended line-mode" ausgegeben.

Das Dienstprogramm CISDICT besteht aus den beiden Moduln DATDIC3 und DATDIC4.

Das Modul DATDIC3 dient der Aufbereitung der Daten im Speicher, DATDIC4 der Ausgabe der Auswertung auf Bildschirm, Drucker oder Datei.

10.2 Auskunftsfunktionen

10.2.1 Auswahl der Auskunftsfunktionen

Im Modul DATDIC4 sind die verschiedenen Auskunftsfunktionen realisiert. Sie sind über eine Bildschirmmaske aufrufbar, die bei Start dieses Moduls und dann jeweils nach Ende der Ausgabe einer Teilfunktion ausgegeben wird. Außer der Auswahl einer Teilfunktion kann auch festgelegt werden, auf welches Medium (Bildschirm, Liste oder Datei) ausgegeben werden soll.

Die Maske hat folgenden Aufbau und Inhalt :

```
                        CIS DD-AUSKUNFT AUS "DABEL"
                        ===========================

UEBERSICHTEN: ALLES       x (Ausgabe: HD-/DB-/Feld-/MSK-Namen)
(Graf.Format) KOMPR.M.KEY x (Ausgabe: HD-/DB-/MSK-Namen + Key-Name)
              KOMPRIMIERT x (Ausgabe: HD-/DB-/TB-/MSK-Namen)

GEZIELTER ZUGANG UEBER:
  HD-NAME             xxx.............................xxx
  SATZBESCHR.-NAME    xxxxxx       (Eingabe: xxxxxx / xxx* / * / ? )
  TRAFO-/BILDBESCHR.  xxxxxx       (Eingabe: xxxxxx / xxx* / * )
  MASKEN-NAME         xxxxxx       (Eingabe: xxxxxx / xxx* / * )
  FELD-NAME           xxx.....xxx  (Eingabe: xx..xx / xxx* / * )

UEBERSICHT:     ALLE HD's x
(Dateiformat)   EINE HD   xxx.............................xxx
                (HD-Name eingeben)

AUSGABE  AUF BILDSCHIRM x
         AUF LISTE      x
         IN  DATEI      x  (nur bei Übersicht in Dateiformat)
BEENDEN x
-------------------------------------------------------------
```

Eingaben sind in der Maske hinter den kursiv (hell) gestellten Begriffen möglich, sowie nach "BEENDEN". Die jeweilige Eingabelänge ist durch Zeichen(-folgen) "x" evtl. mit zwischenliegenden Punkten dargestellt. Bei HD-Namen und Feldnamen ist dies die maximale Eingabelänge. Für die Teilfunktionen des gezielten Zugangs ist (mit Ausnahme beim HD-Namen) auch eine von rechts verkürzte, mit "*" abgeschlossene Eingabe bzw. nur die Eingabe des Zeichens "*" möglich.

Die Eingabeinformation wird vorrangig auf gewünschte Beendigung überprüft. Bei einer Anforderung wird jeweils nur die erste gewünschte Teilfunktion ab Bildschirmbeginn ausgeführt, gleiches gilt für das gewünschte Ausgabemedium.

Bei Ausgabe auf Bildschirm oder Liste wird vor der gewünschten (Teil-)Ausgabe jeweils die angeforderte Auskunft mit folgender Überschrift protokolliert:

```
UB HD=                              DB=    TB=    MD=    FD=              ZUB
```

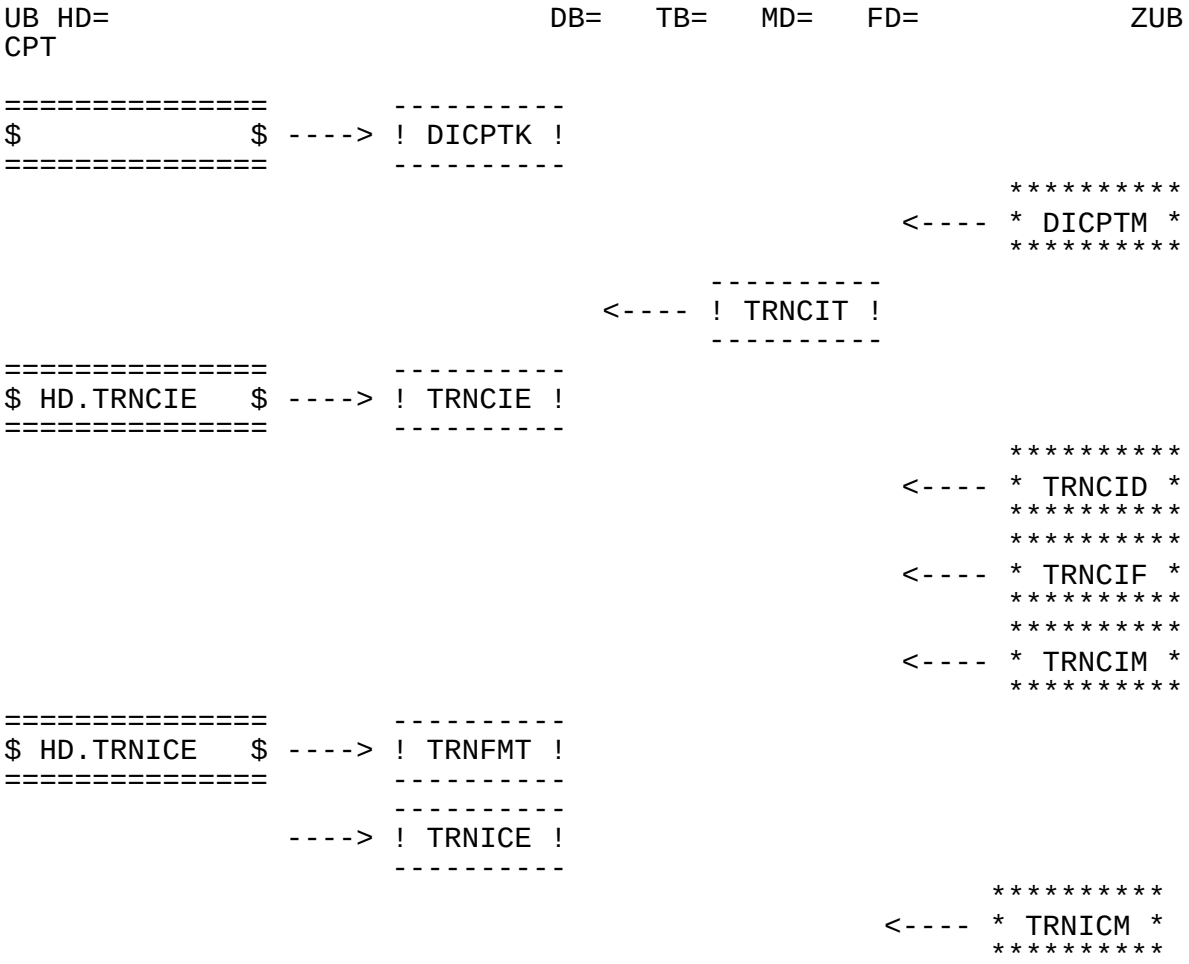
Die jeweils getroffene Auswahl wird in einer weiteren Zeile protokolliert.

10.2.2 Allgemeine Übersichten

Die allgemeine Übersicht kann in drei verschiedenen Arten ausgegeben werden:

Komprimiert	: CPT
Komprimiert mit Key	: CMP
Alles	: ALL

Ausgabeform Komprimiert - CPT



Bedeutung der einzelnen Spalten:

1. Spalte: Darstellung der HD's (mit =/$-Rahmen)
2. Spalte: Satzbeschreibungen (mit -/!-Rahmen)
3. Spalte: Transformations- und Bildbeschreibungen
4. Spalte: Maskenbeschreibungen (mit **-Rahmen)

Der HD-Name wird, wenn sich mehrere Beschreibungen auf die gleiche HD beziehen, nur bei der ersten Satz-, Transformations- oder Bildbeschreibung ausgegeben. Der Name einer Satzbeschreibung wird nur einmal vor der ersten Maskenbeschreibung ausgegeben, selbst wenn mehrere Masken mit der gleichen Satzbeschreibung verbunden sind.

Auf einen "leeren" HD-Namen sind alle Satz-, Transformations- oder Bildbeschreibungen (und auf solche Satzbeschreibungen verweisende Maskenbeschreibungen) zugeordnet, in deren A-Segment kein HD-Parameter enthalten ist.

Ausgabeform Komprimiert mit Key - CMP

Die Ausgabe entspricht der auf Seite 273 dargestellten (CPT) mit dem Unterschied, daß in Spalte 3 der Name des Keyfeldes erscheint.

Ausgabeform Alles - ALL

Die Ausgabe entspricht der auf Seite 273 dargestellten (CPT) mit dem Unterschied, daß in Spalte 3 alle zur jeweiligen Satzbeschreibung gehörenden Feldnamen ohne Rahmen ausgegeben werden.

10.2.3 Gezielte Auswertung über HD

Für die Auswertung über die HD muß ein konkreter HD-Name eingegeben werden. Die Ausgabe bei dieser Teilfunktion hat folgende Form:

```
UB HD=                          DB=    TB=    MD=    FD=             ZUB
   HD.TELEFO

=================          ----------
$ HD.TELEFO     $ -----> ! TELEFO !
=================          ----------

                            LOGADR=J
                            DSS=J
                            DSA=J
                            DSU=J
                            VD=VD.TELEFO
                            MAXDES=20
```

Es wird die Beziehung HD zu Satzbeschreibung (evtl. zu mehreren, die der gleichen HD zugeordnet sind) dargestellt. Zum Satzbeschreibungsnamen werden die sechs wichtigsten Parameter des A-Segments, falls sie vorhanden sind, ausgegeben.

Wird ein "?" statt eines HD-Namens eingegeben, erfolgt in gleicher Weise die Ausgabe aller Satz-, Transformations- und Bildbeschreibungen, die dem "leeren" HD-Namen zugeordnet sind.

10.2.4 Gezielte Auswertung über Satzbeschreibungsnamen

Für die Auswertung über den Satzbeschreibungsnamen kann ein konkreter Name, ein verkürzter mit "*" abgeschlossener Name oder nur "*" eingegeben werden. Dies führt je nach Inhalt der DABEL zur Ausgabe für eine oder mehrere Satzbeschreibungen in folgender Form:

```
UB HD=                          DB=    TB=    MD=    FD=             ZUB
                                TRNICE

                     ----------
SCHLUESSEL     <---- ! TRNICE !
                     ----------

LAUFNR                LOGADR=J
ABFORT                DSS=J
ABFZEIT               DSA=J
ANKORT                DSU=J
ANKZEIT               HD=HD.TRNICE
VTAGE                 VD=VD.TRNICE
                      MAXDES=20
                      EK=
                      /E
                      DB.TRS=N
```

Pro Satzbeschreibung werden alle Parameter des A-Segments, sowie alle zugehörigen Feldnamen in der definierten Reihenfolge dargestellt. Wenn als Trennzeichen bei EK= ein Komma enthalten ist, dann wird dieser Parameter dadurch gesplittet. Das Komma wird nicht mit ausgegeben.

Die Eingabe eines "?" statt eines Satzbeschreibungsnamens bewirkt die Ausgabe aller jener Satzbeschreibungen, die von Maskenbeschreibungen angesprochen werden, aber als Satzbeschreibungen nicht (mehr) vorhanden sind. An Stelle der A-Segment-Parameter wird bei dieser Inkonsistenz der Hinweis "Satzbeschreibung fehlt !!" ausgegeben.

10.2.5 Gezielte Auswertung über die Transformations - oder Bildbeschreibung

Für die Auswertung über den Transformations- oder Bildbeschreibungsnamen kann ein konkreter Name, ein verkürzter mit "*" abgeschlossener Name oder nur "*" eingegeben werden. Je nach Inhalt der DABEL erfolgt für eine oder mehrere Beschreibung(en) die Ausgabe wie im folgenden Beispiel:

```
UB HD=                              DB=    TB=    MD=    FD=              ZUB
                                           TRNCIT

----------
! TRNCIT !
----------

LOGADR=J
DB.TRS=J
```

Pro Beschreibung werden alle Parameter des A-Segments ausgegeben.

10.2.6 Gezielte Auswertung über Maskennamen

Für die Auswertung über den Maskennamen kann ein konkreter Name, ein verkürzter mit "*" abgeschlossener Name oder nur "*" eingegeben werden. Je nach Anforderung und Inhalt der DABEL wird für eine oder mehrere Maskenbeschreibung(en) folgende Information ausgegeben:

```
UB HD=                              DB=    TB=    MD=    FD=              ZUB
                                                  TRNCIM

**********          ----------
* TRNCIM * -----> ! TRNCIE !
**********          ----------

LAUFNR
ABFORT
ABFZEIT
ANKORT
ANKZEIT
VTAGE
```

Pro Maskenbeschreibung erfolgt über die zugeordnete Satzbeschreibung die Ausgabe aller Feldnamen der entsprechenden Satzbeschreibung, die in der Maske angesprochen werden. Die Reihenfolge der Feldnamen entspricht der durch die Maskenbeschreibung gegebenen Reihenfolge.

Wenn sich die Kombination aus Distanz und Länge, die in einer Maskenbeschreibung vorkommt, nicht mit einem Feldnamen und dessen Länge in der Satzbeschreibung deckt, wird für das Maskenfeld der Hinweis "Partner fehlt !" ausgegeben.

10.2.7 Gezielte Auswertung über Feldbezeichnung

Für die Auswertung über den Feldnamen kann ein solcher in voller Länge, ein verkürzter mit "*" abgeschlossener Name oder nur "*" eingegeben werden. Von der letzten Möglichkeit muß jedoch, außer im Fall einer gewollten Gesamtauswertung in Listenform, abgeraten werden, da für alle Felder aller Satzbeschreibungen eine Ausgabe in nachfolgender Form erfolgt.

```
                          ----------
  Feldname        -----> ! DBName !
                          ----------
Distanz: nnnnn Format: xxxxxxxxxxx Log.Bed.: x  Sek.-Typ: xxxxxxxxx
Laenge :   nnn
```

Die Eingabe des Feldnamens ABFORT ergibt beispielsweise folgende Ausgabe:

```
UB HD=                          DB=   TB=   MD=   FD=             ZUB
                                                  ABFORT

                          ----------
  ABFORT          -----> ! TRNCIE !
                          ----------
Distanz: 00010  Format: Zeichen lb              Sek.-Typ: vorhanden
Laenge :   015
                          ----------
  ABFORT          -----> ! TRNFMT !
                          ----------
Distanz: 00010  Format: Zeichen lb              Sek.-Typ: vorhanden
Laenge :   015
                          ----------
  ABFORT          -----> ! TRNICE !
                          ----------
Distanz: 00010  Format: Zeichen lb              Sek.-Typ: vorhanden
Laenge :   015
```

Das Feld ist so oft aufgeführt wie es in den unterschiedlichen Satzbeschreibungen enthalten ist. Im Beispiel sind die Feldcharakteristika in allen Fällen gleich. Sie werden zu Kontrollzwecken bewußt pro Vorkommen ausgegeben, da sie auch unterschiedlich sein könnten.

Die Aussage "Logische Bedeutung" fehlt im Beispiel. Dies bedeutet, daß für die Felddefinitionen keine besondere logische Bedeutung angegeben ist.

Die Eingabe des Feldnamens SCHLUESSEL führt als weiteres Beispiel zu folgender Ausgabe:

```
UB HD=                          DB=    TB=    MD=    FD=              ZUB
                                                      SCHLUESSEL

                          ----------
  SCHLUESSEL      -----> ! TRNCIE !
                          ----------
Distanz: 00005  Format: Binaer     Log.Bed.: O
Laenge :   003
                          ----------
  SCHLUESSEL      -----> ! TRNFMT !
                          ----------
Distanz: 00005  Format: Binaer     Log.Bed.: O
Laenge :   003
                          ----------
  SCHLUESSEL      -----> ! TRNICE !
                          ----------
Distanz: 00005  Format: Binaer     Log.Bed.: O
Laenge :   003
```

Hier fehlt die Aussage "Sek.-Typ". Dies bedeutet, daß für das Feld "SCHLUESSEL" kein Sekundärindex (Verweisdatei) gewünscht ist.

Die auf Seite 277 dargestellte Form der Ausgabe zeigt also den maximalen Umfang der Ausgabeinformation. Die Informationen "Distanz", "Format" und "Laenge" werden stets ausgegeben, die Informationen "Logische Bedeutung" und "Sek.-Typ" nur, wenn sie Teil der jeweiligen Definition sind.

Wenn ein verkürzter Feldname eingegeben wurde, der zu mehreren verschiedenen Feldnamen führt, werden erst alle in der gleichen Satzbeschreibung liegenden dargestellt, erst dann folgt der Wechsel zur nächsten Satzbeschreibung.

10.2.8 Übersicht im Dateiformat

Die Teilfunktion "Übersicht im Dateiformat" kann als einzige auch in eine DVS-Datei ausgegeben werden. Hier unterscheiden sich die Ausgabeformate für Bildschirm/SYSLST und Datei. Die Ausgabe für Bildschirm und SYSLST hat folgendes Aussehen:

```
UB HD=                          DB=    TB=    MD=    FD=             ZUB
                                                                      X

                    ----> DICPTK
                        > DICPTK                              <---- DICPTM
                                    <---- TRNCIT
HD.TRNCIE           ----> TRNCIE VD.TRNCIE
HD.TRNCIE               > TRNCIE                              <---- TRNCID
HD.TRNCIE               > TRNCIE                              <---- TRNCIF
HD.TRNCIE               > TRNCIE                              <---- TRNCIM
HD.TRNICE           ----> TRNFMT VD.TRNICE
HD.TRNICE           ----> TRNICE VD.TRNICE
HD.TRNICE               > TRNICE                              <---- TRNICM
```

Diese Ausgabe kam durch die Anforderung "alle HD's" zustande. Wenn beispielsweise die Anforderung "eine HD" mit dem Namen HD.TRNCIE wäre, würden nur die vier Zeilen ausgegeben, die mit HD.TRNCIE beginnen.

Die ersten drei Zeilen repräsentieren den Fall des "leeren" HD-Namens. HD-Namen und Satzbeschreibungsnamen sind aus Gründen der Sortierbarkeit in allen Zeilen enthalten, denen sie logisch zugehörig sind.

Die einzelnen Spalten beinhalten HD-Name, Satzbeschreibungsname, VD-Name bzw. (wenn mit Linkspfeil beginnend) Transformations-/Bildname und Maskenbeschreibungsname.

Der Rechtspfeil zwischen Spalte 1 und Spalte 2 wird nur mit dem Zeichen ">" dargestellt, wenn in dieser Zeile eine Beziehung Maskenbeschreibung zu Satzbeschreibung vorliegt.

Bei Ausgabe in eine Datei unterscheidet sich das Format vom oben dargestellten durch Wegfall der Pfeile und dadurch, daß HD- und VD-Namen in der maximal möglichen Länge von 54 Stellen ausgegeben werden. Zwischen den einzelnen Spalten (Feldern) der Zeile (des Satzes) liegt jeweils ein Leer-Byte.

Die Ausgabe in Datei ist nur möglich, wenn vor Start des Programms CISDICT mit LINK-NAME=DICOUT eine Datei zugeordnet wurde. Um ein versehentliches Überschreiben dieser Datei zu verhindern ist pro Programmlauf nur einmal die Ausgabe in eine Datei möglich (Datei wird OPEN=OUTPUT eröffnet und nicht explizit geschlossen !).

Die Übersichten in Dateiformat sind als Datei aber auch als SYSLST-Datei mit einer entsprechenden Satzbeschreibung mit CIS.ODASI oder CIS verarbeitbar.